



Trabajo Fin de Grado

SHARECAR. PLATAFORMA SOCIAL PARA LA OPTIMIZACIÓN DE RECURSOS LOGÍSTICOS

Alumno: David Tedejo García

Tutor y co-tutor: José María Álvarez Rodríguez y Lisardo Prieto González

Grado en Ingeniería Informática

Índice de contenido

1. Introducción	9
1.1. Objetivo y alcance.....	9
1.2. Acrónimos y definiciones	10
2. Resumen del proyecto	12
3. Estudio del estado del arte	13
3.1. Valoración del estado del arte	13
3.1.1. Blablacar.es.....	13
3.1.2. Carpooling.es.....	14
3.1.3. Amovens.com.....	14
3.2. Realización de la comparativa del estado del arte	15
4. Objetivos de la aplicación.....	16
5. Metodología, arquitectura y entorno tecnológico	17
5.1. Descripción de la metodología a utilizar	17
5.1.1. Etapas Métrica Versión 3.....	17
5.2. Definición de la arquitectura del sistema	18
5.2.1. Definición de niveles de arquitectura.....	18
5.3. Especificación del entorno tecnológico	19
5.3.1. Hardware.....	19
5.3.1.1. Usuario.....	19
5.3.1.2. Servidor.....	19
5.3.2. Software	19
5.3.2.1. Herramientas empleadas	20
6. Análisis del sistema	21
6.1. Determinación del alcance del sistema.....	21
6.2. Identificación de los usuarios participantes y finales	21
6.3. Especificación de requisitos software	22
6.3.1. Requisitos de usuario.....	24
6.3.2. Requisitos de viaje	26
6.3.3. Requisitos de puntuaciones	28

6.3.4.	Requisitos de comentarios.....	29
6.3.5.	Requisitos de administrador.....	31
6.3.6.	Requisitos de seguridad y validación	32
6.3.7.	Requisitos de interfaz	33
6.3.8.	Requisitos de rendimiento.....	36
6.4.	Especificación de casos de uso	37
7.	Diseño del sistema	51
7.1.	Definición de la arquitectura del sistema	51
7.2.	Requisitos de diseño	51
7.3.	Diseño de clases	54
7.3.1.	Identificación de clases de diseño.....	54
7.4.	Diseño físico de datos	56
7.4.1.	Diseño del modelo físico de datos.....	56
7.4.2.	Especificación de los caminos de acceso a los datos y optimización....	59
7.4.3.	Especificación de la distribución de datos.....	60
8.	Implementación del prototipo	61
8.1.	Estructura del prototipo	61
8.2.	Diseño de las interfaces del prototipo.....	62
9.	Pruebas del sistema.....	70
9.1.	Especificación del plan de pruebas.....	70
9.1.1.	Definición del alcance de las pruebas.....	70
9.1.2.	Definición de requisitos del entorno de pruebas.....	70
9.1.2.1.	Software	71
9.1.2.2.	Hardware	71
9.2.	Especificación técnica de niveles de pruebas.....	72
9.2.1.	Pruebas de integración.....	72
9.2.2.	Pruebas de sistema	80
9.2.2.1.	Pruebas de rendimiento	80
9.2.2.2.	Pruebas de facilidad de uso o aceptación.....	80



9.2.2.3. Pruebas de entorno.....	81
9.2.3. Pruebas de implantación.....	82
10. Conclusiones.....	83
11. Líneas futuras	84
12. Referencias.....	85
13. Anexo I. Presupuesto.....	87
13.1. Presupuesto	87
13.1.1. Personal.....	87
13.1.2. Material.....	91
13.1.2.1. Hardware	91
13.1.2.2. Software.....	91
13.1.3. Costes indirectos.....	92
13.1.4. Resumen de costes	92
13.1.5. Coste total	93
14. Anexo II. Planificación	94
14.1. Planificación inicial	94
14.2. Diagrama de Gantt inicial	96
14.3. Planificación final.....	98
14.4. Diagrama de Gantt final.....	101
14.5. Comparativa planificaciones	103

Índice de ilustraciones

Ilustración 1: Logotipo de BlaBlaCar.....	13
Ilustración 2: Logotipo de Carpooling	14
Ilustración 3: Logotipo de Amovens.....	14
Ilustración 4: Modelo-Vista-Controlador	18
Ilustración 5: Diagrama relaciones de usuarios.....	22
Ilustración 6: Diagrama casos de uso usuario administrador.....	38
Ilustración 7: Diagrama casos de uso usuario no registrado.....	39
Ilustración 8: Diagrama casos de uso usuario registrado	39
Ilustración 9: Diagrama de interacciones entre componentes	55
Ilustración 10: Modelo físico de datos.....	57
Ilustración 11: Estructura prototipo	61
Ilustración 12: Interfaz inicio.....	62
Ilustración 13: Interfaz registro usuarios	63
Ilustración 14: Interfaz inicio sesión.....	64
Ilustración 15: Interfaz perfil de usuario	64
Ilustración 16: Interfaz modificar perfil	65
Ilustración 17: Interfaz viajes del usuario.....	66
Ilustración 18: Interfaz indicar acompañantes	66
Ilustración 19: Interfaz modificar viaje.....	67
Ilustración 20: Interfaz viajes encontrados.....	67
Ilustración 21: Interfaz perfil otro usuario sin iniciar sesión.....	68
Ilustración 22: Interfaz perfil otro usuario iniciando sesión.....	68
Ilustración 23: Interfaz contacto ShareCar	69
Ilustración 24: Diagrama de navegación	69
Ilustración 25: Diagrama de Gantt inicial (1)	97
Ilustración 26: Diagrama de Gantt inicial (2)	97
Ilustración 27: Diagrama de Gantt inicial (3)	98
Ilustración 28: Diagrama de Gantt final (1).....	101
Ilustración 29: Diagrama de Gantt final (2).....	102
Ilustración 30: Diagrama de Gantt final (3).....	102

Índice de tablas

Tabla 1: Comparativa del estado del arte.....	15
Tabla 2: Formato tabla requisitos	23
Tabla 3: Requisito RUSU-001	24
Tabla 4: Requisito RUSU-002	24
Tabla 5: Requisito RUSU-003	24
Tabla 6: Requisito RUSU-004	25
Tabla 7: Requisito RUSU-005	25
Tabla 8: Requisito RUSU-006	25
Tabla 9: Requisito RUSU-007	26
Tabla 10: Requisito RUSU-008	26
Tabla 11: Requisito RVIA-001.....	26
Tabla 12: Requisito RVIA-002.....	27
Tabla 13: Requisito RVIA-003.....	27
Tabla 14: Requisito RVIA-004.....	27
Tabla 15: Requisito RVIA-005.....	28
Tabla 16: Requisito RVIA-006.....	28
Tabla 17: Requisito RPUN-001	28
Tabla 18: Requisito RPUN-002	29
Tabla 19: Requisito RPUN-003	29
Tabla 20: Requisito RCOM-001	29
Tabla 21: Requisito RCOM-002	30
Tabla 22: Requisito RCOM-003	30
Tabla 23: Requisito RCOM-004	30
Tabla 24: Requisito RADM-001	31
Tabla 25: Requisito RADM-002	31
Tabla 26: Requisito RADM-003	31
Tabla 27: Requisito RSEG-001	32
Tabla 28: Requisito RSEG-002	32
Tabla 29: Requisito RSEG-003	32
Tabla 30: Requisito RSEG-004	33
Tabla 31: Requisito RSEG-005	33
Tabla 32: Requisito RINT-001	33
Tabla 33: Requisito RINT-002	34
Tabla 34: Requisito RINT-003	34
Tabla 35: Requisito RINT-004	34
Tabla 36: Requisito RINT-005	35
Tabla 37: Requisito RINT-006	35
Tabla 38: Requisito RINT-007	35
Tabla 39: Requisito RINT-008	36
Tabla 40: Requisito RREN-001	36



Tabla 41: Requisito RREN-002.....	36
Tabla 42: Requisito RREN-003.....	37
Tabla 43: Sistema de tablas para los casos de uso	40
Tabla 44: Caso de uso CU-001.....	40
Tabla 45: Caso de uso CU-002.....	41
Tabla 46: Caso de uso CU-003.....	41
Tabla 47: Caso de uso CU-004.....	41
Tabla 48: Caso de uso CU-005.....	42
Tabla 49: Caso de uso CU-006.....	42
Tabla 50: Caso de uso CU-007.....	42
Tabla 51: Caso de uso CU-008.....	43
Tabla 52: Caso de uso CU-009.....	43
Tabla 53: Caso de uso CU-010.....	43
Tabla 54: Caso de uso CU-011.....	44
Tabla 55: Caso de uso CU-012.....	44
Tabla 56: Caso de uso CU-013.....	44
Tabla 57: Caso de uso CU-014.....	45
Tabla 58: Caso de uso CU-015.....	45
Tabla 59: Caso de uso CU-016.....	45
Tabla 60: Caso de uso CU-017.....	46
Tabla 61: Caso de uso CU-018.....	46
Tabla 62: Caso de uso CU-019.....	46
Tabla 63: Caso de uso CU-020.....	47
Tabla 64: Caso de uso CU-021.....	47
Tabla 65: Caso de uso CU-022.....	47
Tabla 66: Caso de uso CU-023.....	48
Tabla 67: Trazabilidad Requisitos software-Casos de uso	50
Tabla 68: Requisito RDC-001	52
Tabla 69: Requisito RDC-002	52
Tabla 70: Requisito RDC-003	52
Tabla 71: Requisito RDC-004	53
Tabla 72: Requisito RDC-005	53
Tabla 73: Requisito RDC-006	53
Tabla 74: Requisito RDC-007	54
Tabla 75: Tabla usuario.....	57
Tabla 76: Tabla viaje.....	58
Tabla 77: Tabla viajes_realizados.....	58
Tabla 78: Tabla comentarios_usuario.....	59
Tabla 79: Tabla comentarios_web.....	59
Tabla 80: Modelo de tabla REP.....	70
Tabla 81: REP-01.....	71
Tabla 82: REP-02.....	71
Tabla 83: REP-03.....	71



Tabla 84: REP-04.....	71
Tabla 85: REP-05.....	72
Tabla 86: Formato tabla pruebas de integración	72
Tabla 87: Prueba PI-001	73
Tabla 88: Prueba PI-002	73
Tabla 89: Prueba PI-003	74
Tabla 90: Prueba PI-004	74
Tabla 91: Prueba PI-005	75
Tabla 92: Prueba PI-006	75
Tabla 93: Prueba PI-007	76
Tabla 94: Prueba PI-008	76
Tabla 95: Prueba PI-009	77
Tabla 96: Prueba PI-010	77
Tabla 97: Prueba PI-011	78
Tabla 98: Prueba PI-012	78
Tabla 99: Prueba PI-013	79
Tabla 100: Prueba PI-014.....	79
Tabla 101: Prueba PI-015.....	80
Tabla 102: Formato tabla pruebas de entorno	81
Tabla 103: Prueba PSEN-001	82
Tabla 104: Prueba PSEN-002	82
Tabla 105: Estimación de coste de personal por hora	87
Tabla 106: Coste asociado a los recursos y horas empleadas	90
Tabla 107: Coste asociado al <i>hardware</i> empleado.....	91
Tabla 108: Coste asociado al software empleado	92
Tabla 109: Costes indirectos	92
Tabla 110: Resumen de costes	93
Tabla 111: Presupuesto total	93
Tabla 112: Planificación inicial.....	96
Tabla 113: Planificación final.....	100

1. Introducción

Gracias a la enorme expansión de las comunicaciones interpersonales a través de Internet y al continuo desarrollo y aparición de nuevos dispositivos, tanto de escritorio como móviles con acceso a redes de datos, han emergido numerosos servicios con objetivo de mejorar la calidad de vida y prestar servicios que sin dichas tecnologías serían imposibles de prestar. Entre ellos se enmarca el presente proyecto. En el mismo, se pretende desarrollar un sistema que tiene como objetivo la optimización de los recursos de las personas mediante un sistema de recomendación social. Dichos recursos son los vehículos que usan estas personas, y la finalidad del proyecto es que personas que realicen viajes similares puedan aprovechar estos recursos de una forma más óptima compartiéndolos.

A lo largo del documento se define el proyecto software a desarrollar, para ello se ha seguido la metodología Métrica V3 [1] con algunos puntos adicionales. Primero se definen 1.1 Objetivo y alcance, donde se explica cuál es el objetivo del presente documento y el alcance que tiene el mismo, y 1.2 Acrónimos y definiciones, donde se encuentran todas las palabras o acrónimos que puedan conllevar a confusión del presente documento. Tras esto se realiza un pequeño resumen en la sección 2. Resumen del proyecto.

Luego se empieza con las pautas que marca Métrica V3, las cuales incluyen las secciones 3. Estudio del estado del arte y 4. Objetivos de la aplicación, que sería el Estudio de Viabilidad del Sistema, 5. Metodología, arquitectura y entorno tecnológico, definiendo cómo se va a desarrollar el sistema, 6. Análisis del sistema, 7. Diseño del sistema, 8. Implementación del prototipo, que se puede entender como Implantación y Aceptación del Sistema y 9. Pruebas del sistema

Por último, se encuentran las secciones de 10. Conclusiones, 11. Líneas futuras y 12. Referencias, también se incluyen dos anexos donde se encuentra el presupuesto del proyecto, 13. Anexo I. Presupuesto y la planificación del mismo, 14. Anexo II. Planificación.

1.1. Objetivo y alcance

El objetivo de este documento es realizar un seguimiento de todas las etapas de un proyecto, desde que se tiene la idea hasta la realización (en este caso codificación) de ésta. Para ello se han seguido las pautas que establece Métrica V3 en la realización de proyectos, las cuales a grandes rasgos se pueden resumir en:

- Estudio de viabilidad del sistema o estudio del arte
- Análisis del sistema
- Diseño del sistema
- Pruebas del sistema

Y para que el proyecto sea más completo, se han añadido otros puntos como son

- Resumen del proyecto
- Conclusiones
- Líneas futuras

1.2. Acrónimos y definiciones

En esta sección se identifican los acrónimos y definiciones de palabras que aparecen en el presente documento y que pueden generar confusión, para definirlos se han consultado el diccionario de la Real Academia Española [2], entradas de Wikipedia [3], siempre y cuando tuviera referencias serias respecto al contenido y páginas web oficiales como la de *Apache HTTP Server* [4], *MySQL Community Edition* [5], *BootStrap* [6], *BlaBlaCar* [7], *Carpooling* [8] o *Amovens* [9].

- **Amovens:** aplicación web que se dedica a poner en contacto a usuarios para que éstos compartan vehículo en sus viajes.
- **Android:** sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles con pantalla táctil como teléfonos inteligentes o tabletas inicialmente desarrollados por Android Inc.
- **Apache HTTP Server:** servidor web HTTP de código abierto que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.
- **ASI:** siglas de Análisis del Sistema de Información, es una parte del proyecto siguiendo los estándares de Métrica V3.
- **BlaBlaCar:** aplicación web que se dedica a poner en contacto a usuarios que desean compartir vehículo en sus viajes.
- **BootStrap:** conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.
- **Carpooling:** aplicación web que pone en contacto a usuarios que quieran compartir vehículo en sus viajes.
- **CSS:** siglas de *Cascading Style Sheets* (“hojas de estilo en cascada”), es un lenguaje de programación utilizado para describir el aspecto y el formato de un documento escrito en un lenguaje de marcas.
- **DSI:** siglas de Diseño del Sistema de Información, es una parte del proyecto siguiendo los estándares de Métrica V3.
- **EVS:** siglas de Estudio de Viabilidad del Sistema, es una parte del proyecto siguiendo los estándares de Métrica V3.
- **Facebook:** aplicación web de redes sociales creado por Mark Zuckerberg y fundado junto a Eduardo Saverin, Chris Hughes y Dustin Moskovitz. Originalmente era un sitio para estudiantes de la Universidad de Harvard, pero se abrió a cualquier persona con una cuenta de correo electrónico.
- **HTML:** siglas de *HyperText Markup Language* (“lenguaje de marcas de hipertexto”), es un lenguaje de programación que se utiliza para la elaboración de páginas web.
- **HTTP:** siglas de *HyperText Transfer Protocol* (“protocolo de transferencia de hipertexto”), es un protocolo usado en cada transacción realizada en la web que sigue el esquema petición-respuesta entre un cliente y un servidor.
- **IAS:** siglas de Implantación y Aceptación del Sistema, es una parte del proyecto siguiendo los estándares de Métrica V3.
- **iOS:** sistema operativo móvil de la empresa Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV.
- **JavaScript:** lenguaje de programación orientado a objetos, basado en prototipos, imperativo y dinámico. Sirve de dar apoyo a las páginas web capturando eventos y generando acciones.

- **jQuery:** biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos y desarrollar animaciones.
- **Métrica V3:** metodología de planificación, desarrollo y mantenimiento de sistemas de información, promovida por el Ministerio de Administraciones Públicas del Gobierno de España para la sistematización de actividades del ciclo de vida de los proyectos software en el ámbito de las administraciones públicas.
- **MVC:** siglas de Modelo-Vista-Controlador, es un patrón de arquitectura de software que separa los datos y la lógica del negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.
- **MySQL Community Edition:** sistema de gestión de bases de datos relacional, multihilo y multiusuario.
- **PHP:** lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web.
- **PSI:** siglas de Planificación de Sistema de Información, es una parte del proyecto siguiendo los estándares de Métrica V3.
- **Responsive web design:** filosofía de diseño y desarrollo web que mediante el uso de estructuras e imágenes fluidas consigue adaptar el sitio web al entorno del usuario.
- **ShareCar:** nombre de la aplicación web que se ha creado para el desarrollo de este proyecto.
- **Stakeholders:** persona o entidad que se ve afectada por las actividades de una empresa.
- **URL:** siglas de *Uniform Resource Locator* ("Localizador de Recursos Uniformes"), es una cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en Internet.
- **Wikipedia:** proyecto de enciclopedia web multilingüe de contenido libre basado en un modelo de edición abierta.

2. Resumen del proyecto

Este proyecto tiene por objetivo la creación de una aplicación web que principalmente tiene como finalidad la compartición de vehículos entre personas que trabajan o estudian cerca, buscando con ello la optimización de sus recursos mediante un sistema de recomendación social.

En otras palabras, se va a desarrollar una aplicación enfocada a que, a través de recomendaciones de perfiles, personas que trabajen o estudien cerca y usen medio de transporte propio puedan compartir vehículos ahorrando dinero e incrementado su vida social de una forma positiva.

El nombre elegido para la aplicación es **ShareCar**, un bastante fácil de recordar e intuitivo, ya que en inglés significa “compartir coche”.

La aplicación se puede usar sin necesidad de estar registrado, se pueden buscar viajes y ver los perfiles de otros usuarios registrados. Pero para una utilización óptima se debe estar registrado, ya que en caso contrario no se facilita la información necesaria para contactar con los demás usuarios de la aplicación. Igualmente, al estar registrado también se puede puntuar y comentar acerca de otros usuarios, o incluso sobre la propia web **ShareCar**, con el objetivo de proporcionar *feedback* a otros usuarios del sistema o incluso al desarrollador del mismo.

Por último, decir que, en líneas generales, la aplicación se basa en un buscador de viajes creados por los propios usuarios, y que tiene la opción de recomendación de perfiles. Este elemento es muy positivo, ya que ninguna otra aplicación competidora en el mercado presenta dicha funcionalidad.

3. Estudio del estado del arte

En esta sección se especifica el estado del arte actual. Es decir, cuál es el estado de la situación actual de otros sistemas existentes similares al que se pretende desarrollar mediante este proyecto.

El objetivo de esta sección es realizar una valoración de la información inferida acerca de los sistemas existentes. Así pues, como resultado de esta actividad se genera un diagnóstico, estimando la eficiencia de cada uno de los sistemas existentes que se han estudiado, identificando los problemas que presentan, proponiendo mejoras y funcionalidades adicionales carentes en los mismos.

3.1. Valoración del estado del arte

A lo largo de esta sección se presentan los sistemas analizados junto con sus principales características.

3.1.1. Blablacar.es



Ilustración 1: Logotipo de BlaBlaCar

BlaBlaCar es una aplicación web que está destinada a poner en contacto a personas que buscan una forma de ahorrar dinero compartiendo coche para sus viajes. Es una aplicación que fomenta la sociabilidad al permitir que personas que no se conocen interactúen y da la oportunidad de conocerse a dichas personas.

Las principales características que ofrece BlaBlaCar son:

- Buscador de viajes introduciendo el origen, el destino y la fecha.
- Creación de alertas que se notifican en el correo personal cuando hay nuevos viajes.
- Posibilidad de sincronizar la cuenta de BlaBlaCar con la cuenta de Facebook.
- Opción multilinguaje, que cuenta con diez idiomas distintos.
- Posibilidad de valorar a los usuarios que utilizan esta aplicación y comentar sobre ellos.
- Diseño sencillo, práctico y fácil de usar.
- Es una aplicación gratuita para todos sus usuarios.
- Aplicación móvil para Android e iOS.

3.1.2. Carpooling.es



Ilustración 2: Logotipo de Carpooling

Carpooling es una aplicación web similar a BlaBlaCar, proporciona facilidad a la hora de encontrar a una persona para compartir un viaje y ahorrar en gastos para el conductor y en billetes de cualquier medio de transporte para los acompañantes.

La interfaz de esta aplicación da una primera impresión de estar más orientada a toda Europa que BlaBlaCar. Algunas de sus características son:

- Es una aplicación gratuita para todos sus usuarios.
- Buscador de viajes introduciendo el origen, el destino y la fecha, con gran flexibilidad a la hora de elegir el radio de kilómetros del origen y del destino y que la fecha pueda ser uno o varios días antes o después de la elegida.
- Creación de alertas que se notifican en el correo personal cuando haya nuevos viajes disponibles.
- Creación de peticiones de viajes.
- Posibilidad de sincronizar la cuenta de Carpooling con la cuenta de Facebook.
- Opción multilinguaje, que cuenta con nueve idiomas distintos.
- Posibilidad de valorar a los usuarios que utilizan esta aplicación y comentar sobre ellos.
- Diseño sencillo y fácil de usar.
- Cuando no se encuentran viajes dan información de qué poder hacer para encontrarlos.
- Aplicación móvil para Android e iOS.

3.1.3. Amovens.com



Ilustración 3: Logotipo de Amovens

Amovens es una aplicación española que se dedica a poner en contacto a conductores y pasajeros que viajan a un mismo lugar, para que así que puedan compartir coche y ahorrar gastos en todo tipo de viajes.



Sus principales características, todas muy parecidas a sus competidores son:

- Es una aplicación gratuita para todos sus usuarios.
- Buscador de viajes introduciendo el origen, el destino y la fecha.
- Chat de contacto disponible en todo momento para dudas o incidencias.
- Creación de alertas que se notifican en el correo personal cuando haya nuevos viajes disponibles.

- Creación de peticiones de viajes.
- Posibilidad de valorar a los usuarios que utilizan esta aplicación y comentar sobre ellos.
- Posibilidad de sincronizar la cuenta de Amovens con la cuenta de Facebook.
- Opción multilinguaje, que cuenta con cinco idiomas distintos.
- Diseño sencillo, práctico y fácil de usar.
- Aplicación móvil para Android e iOS.

3.2. Realización de la comparativa del estado del arte

En esta sección se indicarán qué características posee cada uno de los distintos sistemas mencionados en la sección anterior.

A continuación se incluye una comparativa con las características de los diferentes sistemas de información evaluados. Intuitivamente se ha empleado el símbolo  para indicar que el citado sistema cumple con esa característica, mientras que el símbolo  significa que no la cumple.



Características			
Gratuito			
Multilinguaje			
Buscador de viajes			
Alertas para viajes			
Peticiones de viajes			
Valoración de usuarios			
Comentarios sobre usuarios			

Tabla 1: Comparativa del estado del arte

De los resultados obtenidos se analiza que características se cree que son más importantes y se decide que se va a desarrollar un sistema que cumpla la mayoría de las características que tienen estos sistemas, que podrá solucionar los problemas y características que éstos no cumplen y que incluirá nuevas características.

4. Objetivos de la aplicación

Se va a desarrollar una aplicación web gratuita con la finalidad de que personas que no se conocen puedan contactar entre sí para compartir coche en viajes y así ahorrar dinero en estos desplazamientos.

El tipo de viajes que se van a contemplar son tanto los viajes puntuales de un sitio a otro como los viajes periódicos, es decir, por ejemplo para personas que trabajen o estudien en lugares cercanos y que viajen desde un sitio próximo con regularidad.

La aplicación contará con un servicio de alertas para los usuarios cuando salgan nuevos viajes que están buscando y también se podrán crear peticiones de viajes por si algún otro usuario se quiere ofrecer a hacerlos.

Los usuarios registrados podrán valorar y poner comentarios a los conductores y a los pasajeros con los que hayan viajado anteriormente para facilitar a los demás usuarios información sobre éstos.

En relación a nuevas características, la mayor disrupción que tendrá esta aplicación comparándola con las ya existentes es que se llevará a cabo una recomendación de usuarios. Con una comparación de los perfiles que tienen los usuarios registrados se indicará a cada uno los posibles usuarios con los que se podría identificar mejor y así hacer el proceso más cómodo.

5. Metodología, arquitectura y entorno tecnológico

En esta sección se explica la metodología empleada para el desarrollo del proyecto, la arquitectura que se ha seguido durante la creación del sistema y tanto el software como el hardware usado en el entorno tecnológico.

5.1. Descripción de la metodología a utilizar

La metodología utilizada en la elaboración del presente proyecto es Métrica V3, la cual ha sido desarrollada por el Ministerio de Administraciones Públicas. Para la realización del proyecto se han seguido las etapas que utiliza Métrica V3 y que se explican más adelante.

5.1.1. Etapas Métrica Versión 3

- 1. Planificación de Sistemas de Información (PSI).** Consiste en analizar las necesidades de los sistemas de información que tiene la organización.
- 2. Desarrollo de Sistemas de Información.** Está subdividido en los siguientes procesos:

- **Estudio de Viabilidad del Sistema (EVS)**

El objetivo del Estudio de Viabilidad del Sistema es el análisis de un conjunto concreto de necesidades para proponer una solución a corto plazo, que tenga en cuenta restricciones económicas, técnicas, legales y operativas. La solución obtenida como resultado del estudio, puede ser la definición de uno o varios proyectos que afecten a uno o varios sistemas de información ya existentes o nuevos. Para ello, se identifican los requisitos que se ha de satisfacer y se estudia, si procede, la situación actual. En este proyecto se le ha llamado estudio del estado del arte.

- **Análisis del Sistema de Información (ASI)**

El objetivo de este proceso es la obtención de una especificación detallada del sistema de información que satisfaga las necesidades de información de los usuarios y sirva de base para el posterior diseño del sistema.

- **Diseño del Sistema de Información (DSI)**

El objetivo del proceso de Diseño del Sistema de Información (DSI) es la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.

A partir de dicha información, se generan todas las especificaciones de construcción relativas al propio sistema, así como la descripción técnica del plan de pruebas, la definición de los requisitos de implantación y el diseño de los procedimientos de migración y carga inicial, éstos últimos cuando procedan.

- **Implantación y Aceptación del Sistema (IAS)**

Por último, se realiza la implantación y aceptación del sistema, donde se comprobará la instalación y el funcionamiento del producto con el fin de comprobar si el producto está acabado o es necesario que se realicen más pruebas y se hagan más cambios. Para llevar a cabo esta fase se toman como elementos de partida los productos obtenidos en la fase de DSI.

Así, las sub-tareas más importantes en este punto serán la comprobación del sistema donde se va a realizar la implementación, la instalación del producto y las pruebas realizadas con el producto.

3. Mantenimiento del Sistema de Información

El objetivo de este proceso es la obtención de una nueva versión de un sistema de información desarrollado con Métrica V3, a partir de las peticiones de mantenimiento que los usuarios realizan con motivo de un problema detectado en el sistema, o por la necesidad de una mejora del mismo.

5.2. Definición de la arquitectura del sistema

5.2.1. Definición de niveles de arquitectura

En esta sección se define a alto nivel las estructuras de agrupamiento de los distintos componentes de nuestro sistema. Cada agrupación contendrá las partes significativas del sistema que tienen características propias de función o ejecución.

Para la realización de esta agrupación se utilizará el Modelo-Vista-Controlador (MVC), dado que se considera que es la más eficiente para un sistema como este en el que el usuario accederá desde un navegador web.

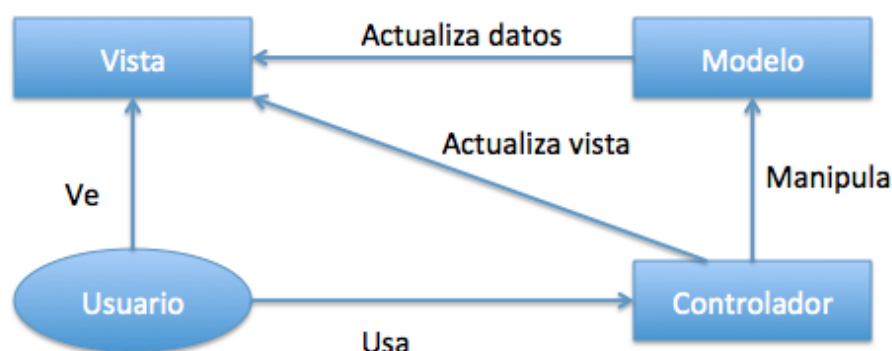


Ilustración 4: Modelo-Vista-Controlador

En ella existen tres niveles de arquitectura:

- **Vista:** presenta el 'Modelo' en un formato adecuado para interactuar con el usuario, requiere la información que debe presentar como salida de dicho 'Modelo'.
- **Modelo:** es la representación de la información con la cual el sistema opera, gestiona todos los accesos a dicha información y los privilegios de éstos. Envía a la 'Vista' información para que sea mostrada y recibe peticiones a través del 'Controlador'.
- **Controlador:** responde a eventos del usuario y envía peticiones al 'Modelo' cuando se solicita información. También puede enviar comandos a la 'Vista' para cambiar la forma en que se presenta el 'Modelo'. Se puede decir que actúa de intermediario entre la 'Vista' y el 'Modelo'.

5.3. Especificación del entorno tecnológico

En esta sección se definen y explican las necesidades tecnológicas que se requieren para el correcto funcionamiento del sistema. Además de esto, también se incluye una descripción de las herramientas empleadas.

5.3.1. Hardware

En esta sección se define, a nivel físico, todo lo necesario para el funcionamiento del sistema.

5.3.1.1. Usuario

A nivel de usuario que ejecute la web, simplemente será necesario tener instalada la última versión de los navegadores web soportados (Google Chrome [10], Safari [11] o Firefox [12]) desde los cuales acceder a la web del sistema.

5.3.1.2. Servidor

En cuanto a los componentes físicos necesarios en el lado de la aplicación necesitaremos un servidor dónde se alojarán la base de datos, el sistema operativo servidor y todo el contenido web ejecutado mediante Apache HTTP Server.

Además de esto, sería necesario un soporte de almacenamiento suficiente como para contener toda la información relativa a los datos de los usuarios y sus viajes.

5.3.2. Software

En esta sección se especifica qué configuración ha sido elegida para el desarrollo del proyecto.

Se ha decidido que el lenguaje de programación que se utilizará es HTML, JavaScript, PHP, CSS para el contenido web. El uso de estos lenguajes se debe a las grandes ventajas que tienen como son: su portabilidad, eficiencia, facilidad de desarrollo, estabilidad o que la existencia de bibliotecas ya implementadas [13].

El servidor web que se utilizará será *Apache HTTP Server*, que además de ser un servidor de código abierto, se proporciona de manera gratuita, consta de una fuerte comunidad de soporte y desarrollo y es una solución probada y ampliamente extendida. De la misma manera ocurre con el gestor de base de datos elegido, *MySQL Community Edition*.

5.3.2.1. Herramientas empleadas

A continuación se definirán las herramientas que han sido empleadas para la elaboración del sistema.

- **Editor de texto** para la programación de la aplicación.
- ***MySQL Community Edition*** para la generación de la base de datos.
- **Microsoft Office** para la realización de la documentación del sistema.
- **Navegadores web** para las pruebas del sistema, se han usado Google Chrome, Safari y Firefox.
- **Sistema Operativo *Mac OS X Mavericks***, para la ejecución de todo lo anterior.
- **Sistema Operativo Windows 8**, para la ejecución de *Microsoft Project 2013*.

Cabe destacar que, además de estas aplicaciones, se han usado algunas de menor envergadura pero solamente se listan las anteriores debido a su importancia en el desarrollo del proyecto.

6. Análisis del sistema

Esta sección tiene como objetivo obtener una especificación detallada del sistema que se va a construir. Asimismo, el análisis sirve como base para el posterior diseño del sistema.

Se determina el alcance del sistema, se identifica a los usuarios participantes y finales que se ven afectados con el proyecto y se especifican tanto los requisitos software como los casos de uso del sistema.

6.1. Determinación del alcance del sistema

El producto que se va a describir a continuación se basa en conseguir, de una manera más fácil y barata, viajar y compartir vehículo en desplazamientos por la ciudad, ya sean cotidianos o esporádicos.

Esto se ofrece a través del portal web **ShareCar**, el cual es accesible por todo el que tenga intención de realizar lo descrito en el párrafo anterior y tenga acceso a Internet mediante un navegador web.

Con **ShareCar** cada usuario puede acceder a la web y buscar viajes que satisfagan sus necesidades para luego ponerse en contacto con el conductor. Por otro lado, los usuarios registrados tienen la opción de crear viajes que vayan a realizar como conductores.

Estos usuarios pueden ver otros perfiles de personas con gustos similares gracias a la recomendación de perfiles de **ShareCar** y, además, valorar tanto a otros usuarios de la plataforma como a la propia página web.

La funcionalidad completa se definirá a lo largo del presente documento.

Los requisitos de usuario identificados en el análisis del sistema se aplicarán al proyecto **ShareCar**. Estos requisitos deberán ser tenidos en cuenta durante el resto de fases del ciclo de vida del proyecto, ya que representan las bases sobre las que se desarrolla el sistema.

6.2. Identificación de los usuarios participantes y finales

Los *stakeholders* son todas las personas u organizaciones que se ven afectas en cualquier punto relacionado con el sistema y su elaboración. Los intereses de cada uno pueden ser diferentes, por ejemplo, los usuarios buscan un sistema que cumpla sus expectativas y los desarrolladores buscan obtener la productividad necesaria para el desarrollo del proyecto.

Los distintos interesados en el sistema son:

- **Clientes:** Son las personas que pretenden obtener algún tipo de beneficio del sistema, por ejemplo económico, tras el desarrollo del mismo.
- **Usuarios:** Son las personas que, una vez desarrollado el sistema, interactúan con él y lo usan para satisfacer sus necesidades.

- **David Tedejo García:** alumno encargado de realizar el proyecto en concreto y de desarrollar la aplicación.
- **Tutor y co-tutor:** personas encargadas de guiar al alumno en el desarrollo del proyecto.
- **Patrocinadores:** Las personas o grupos que puedan proveer de recursos financieros para la posterior comercialización del proyecto.
- **Competencia:** Son las empresas del mismo sector que pueden verse influidas por la creación del sistema.

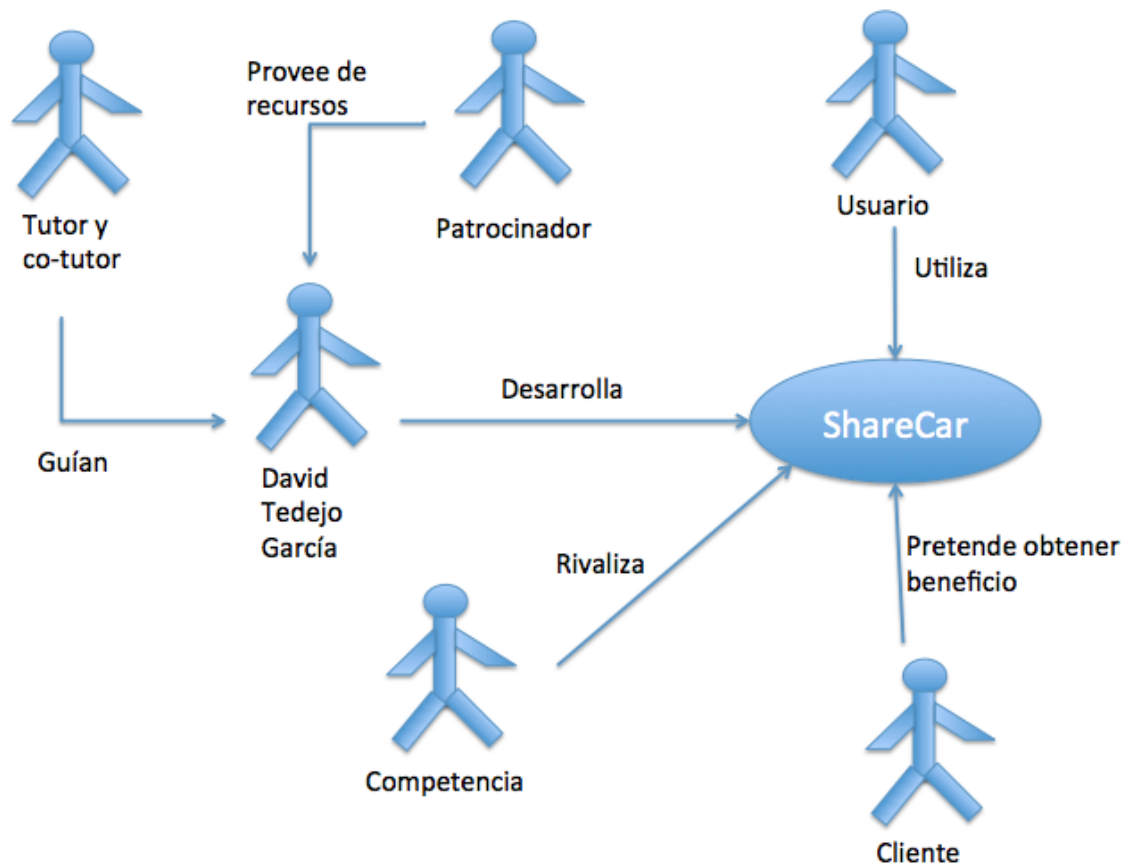


Ilustración 5: Diagrama relaciones de usuarios

6.3. Especificación de requisitos software

En esta sección se realizará la obtención de los requisitos software del sistema. Estos requisitos definirán qué debe hacer el sistema sin preocuparse del cómo debe hacerse, punto que se tratará en la etapa de diseño.

Para que la obtención de requisitos de software fuese lo más completa posible se ha decidido dividirlos en pequeñas subcategorías de manera que se encontrarán definidas todas las partes del sistema. Las categorías para estos requisitos son:

- **Usuario:** Hace referencia a todos aquellos requisitos que tienen que ver con la gestión de usuarios en el sistema, tanto si están registrados como si no.
- **Viaje:** Especifican las funcionalidades de cada viaje dentro del sistema.
- **Puntuaciones:** Especifican las funcionalidades que se ofrecen en relación a las puntuaciones que se otorgan unos usuarios a otros.

- **Comentarios:** Especifican las funcionalidades que se ofrecen en relación a los comentarios de unos usuarios a otros o a la página web.
- **Administrador:** Hace referencia a todos aquellos requisitos que tienen que ver con la gestión del sistema por el administrador.
- **Seguridad y validaciones:** Especifican las capacidades de seguridad que debe ofrecer el sistema y delimita la validación de los campos críticos de la aplicación.
- **Interfaz:** Especifica las funcionalidades que debe ofrecer la interfaz del sistema.
- **Rendimiento:** Especifica el rendimiento mínimo que debe cumplir el sistema.

Una vez identificadas las categorías, se utilizará una tabla para normalizar la definición de los diferentes requisitos. Dicha tabla será mostrada a continuación y tendrá los siguientes campos:

ID				Categoría			
Nombre				Fuente			
Descripción							
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 2: Formato tabla requisitos

- **ID:** Identificador unívoco de cada requisito que tendrá el siguiente formato: **RXXX-NNN**, donde **R** identifica que es un requisito, **XXX** son las tres primeras iniciales de la categoría del requisito y **NNN** es un número de tres cifras diferenciador para cada requisito.
- **Categoría:** Define el tipo de requisito del que se trata.
- **Nombre:** Hace referencia a un pequeño título aclarativo y sencillo de lo que trata el requisito.
- **Fuente:** Es el origen de cada requisito.
- **Descripción:** Representa la definición completa y detallada de cada requisito.
- **Prioridad:** Es la necesidad de que un requisito se realice frente a otro, cuanto más prioridad, antes debe ser realizado.
- **Necesidad:** Es el interés de que un requisito sea realizado. Si es **necesario** tendrá que realizarse en cualquier caso, si es **deseable** se intentará realizar pero podrá ser eliminado por falta de tiempo y si es **opcional** se realizará solo en caso de que el resto de requisitos se cumplan por completo.
- **Estabilidad:** Es la probabilidad de que un requisito sea modificado. Cuanto mayor sea su estabilidad más difícil es que cambie.
- **Riesgo:** Es la necesidad del requisito dentro del sistema acompañado de la dificultad de su desarrollo. Es decir, un requisito con riesgo bajo tiene poco impacto sobre el resto del sistema y es fácil de desarrollar mientras que un requisito con riesgo alto tiene un gran impacto sobre el funcionamiento del sistema y puede ser complejo en su desarrollo.

- **Verificabilidad:** Indica si es posible comprobar que el requisito ha sido añadido al diseño.
- **Claridad:** Es la facilidad que tiene el requisito haciendo referencia a su entendimiento.

6.3.1. Requisitos de usuario

ID	RUSU-001			Categoría	Usuario		
Nombre	Registrar usuario			Fuente	David Tedejo García		
Descripción	El usuario será registrado en el sistema siempre que cumpla con las necesidades requeridas.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 3: Requisito RUSU-001

ID	RUSU-002			Categoría	Usuario		
Nombre	Borrar usuario			Fuente	David Tedejo García		
Descripción	Un usuario registrado que desee abandonar de forma permanente el sistema será borrado del mismo.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 4: Requisito RUSU-002

ID	RUSU-003			Categoría	Usuario		
Nombre	Modificar usuario			Fuente	David Tedejo García		
Descripción	Un usuario registrado podrá modificar los datos personales que quiera de su perfil. Para ello deberá haber accedido al sistema previamente.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 5: Requisito RUSU-003

ID	RUSU-004			Categoría	Usuario		
Nombre	Inicio sesión			Fuente	David Tedejo García		
Descripción	Un usuario registrado podrá acceder al sistema mediante el inicio de sesión siempre que se haya registrado previamente.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 6: Requisito RUSU-004

ID	RUSU-005			Categoría	Usuario		
Nombre	Cierre sesión			Fuente	David Tedejo García		
Descripción	Un usuario registrado podrá cerrar su sesión dentro del sistema de manera que lo abandonará de forma temporal. Para ello deberá haber accedido al sistema previamente.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 7: Requisito RUSU-005

ID	RUSU-006			Categoría	Usuario		
Nombre	Ver perfil			Fuente	David Tedejo García		
Descripción	Un usuario registrado podrá visualizar su perfil a través de la interfaz del sistema.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 8: Requisito RUSU-006

ID	RUSU-007			Categoría	Usuario		
Nombre	Modificar contraseña			Fuente	David Tedejo García		
Descripción	Un usuario registrado podrá modificar la contraseña de su cuenta. Para ello deberá haber accedido al sistema previamente.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 9: Requisito RUSU-007

ID	RUSU-008			Categoría	Usuario		
Nombre	Ver perfiles similares			Fuente	David Tedejo García		
Descripción	Un usuario registrado puede ver perfiles similares al suyos mediante la recomendación de perfiles de ShareCar .						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 10: Requisito RUSU-008

6.3.2. Requisitos de viaje

ID	RVIA-001			Categoría	Viaje		
Nombre	Buscar viaje			Fuente	David Tedejo García		
Descripción	Cualquier usuario de la aplicación puede buscar viajes que se amolden a sus necesidades.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 11: Requisito RVIA-001

ID	RVIA-002			Categoría	Viaje		
Nombre	Crear viaje			Fuente	David Tedejo García		
Descripción	Un usuario registrado puede crear viajes que vaya a realizar con el fin de buscar acompañantes.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 12: Requisito RVIA-002

ID	RVIA-003			Categoría	Viaje		
Nombre	Borrar viaje			Fuente	David Tedejo García		
Descripción	Un usuario registrado puede cancelar (borrar) un viaje que tenía previsto y no puede realizar.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 13: Requisito RVIA-003

ID	RVIA-004			Categoría	Viaje		
Nombre	Modificar viaje			Fuente	David Tedejo García		
Descripción	Un usuario registrado puede modificar los datos de un viaje que tiene previsto realizar.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 14: Requisito RVIA-004

ID	RVIA-005			Categoría	Viaje		
Nombre	Ver viajes realizados			Fuente	David Tedejo García		
Descripción	Un usuario registrado debe poder ver los viajes que ya ha realizado.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 15: Requisito RVIA-005

ID	RVIA-006			Categoría	Viaje		
Nombre	Ver viajes disponibles			Fuente	David Tedejo García		
Descripción	Cualquier usuario puede ver viajes disponibles en la interfaz de inicio de la aplicación.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 16: Requisito RVIA-006

6.3.3. Requisitos de puntuaciones

ID	RPUN-001			Categoría	Puntuaciones		
Nombre	Ver puntuación usuario			Fuente	David Tedejo García		
Descripción	Un usuario registrado puede ver la puntuación de otro o la suya propia a través de la interfaz de perfil de usuario.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 17: Requisito RPUN-001

ID	RPUN-002			Categoría	Puntuaciones		
Nombre	Puntuar usuario			Fuente	David Tedejo García		
Descripción	Un usuario registrado puede puntuar a otro a través de la interfaz de perfil del otro usuario.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 18: Requisito RPUN-002

ID	RPUN-003			Categoría	Puntuaciones		
Nombre	Ver usuarios mejor puntuados			Fuente	David Tedejo García		
Descripción	Cualquier usuario puede ver los usuarios mejor puntuados en la interfaz de inicio de ShareCar .						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 19: Requisito RPUN-003

6.3.4. Requisitos de comentarios

ID	RCOM-001			Categoría	Comentarios		
Nombre	Ver comentarios a usuarios			Fuente	David Tedejo García		
Descripción	Un usuario puede ver los comentarios realizados sobre otros usuarios o sobre él mismo en la interfaz de perfil de usuario.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 20: Requisito RCOM-001

ID	RCOM-002			Categoría	Comentarios		
Nombre	Ver comentarios ShareCar			Fuente	David Tedejo García		
Descripción	Un usuario puede ver los comentarios realizados sobre la página web en la interfaz de contacto de ShareCar .						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 21: Requisito RCOM-002

ID	RCOM-003			Categoría	Comentarios		
Nombre	Crear comentario sobre otro usuario			Fuente	David Tedejo García		
Descripción	Un usuario registrado puede crear un comentario sobre otro usuario a través de la interfaz de perfil de este último.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 22: Requisito RCOM-003

ID	RCOM-004			Categoría	Comentarios		
Nombre	Crear comentario sobre ShareCar			Fuente	David Tedejo García		
Descripción	Un usuario registrado puede crear un comentario sobre ShareCar a través de su interfaz de perfil.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 23: Requisito RCOM-004

6.3.5. Requisitos de administrador

ID	RADM-001			Categoría	Administrador		
Nombre	Denegación acceso			Fuente	David Tedejo García		
Descripción	El administrador puede denegar el acceso al sistema a otros usuarios por motivos de seguridad o de indisciplina.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 24: Requisito RADM-001

ID	RADM-002			Categoría	Administrador		
Nombre	Ver estado de los servidores			Fuente	David Tedejo García		
Descripción	El administrador del sistema puede obtener la información y comprobar periódicamente el estado de los servidores de almacenamiento.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 25: Requisito RADM-002

ID	RADM-003			Categoría	Administrador		
Nombre	Administración del sistema			Fuente	David Tedejo García		
Descripción	El administrador del sistema se encarga de realizar todas las funciones de administración del sistema, ya sea organizar, aplicar cambios, etc.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 26: Requisito RADM-003

6.3.6. Requisitos de seguridad y validación

ID	RSEG-001			Categoría	Seguridad		
Nombre	Ocultar información de perfil de usuario			Fuente	David Tedejo García		
Descripción	El sistema ocultará información de contacto del usuario a otros usuarios que no estén registrados.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 27: Requisito RSEG-001

ID	RSEG-002			Categoría	Seguridad		
Nombre	Control de salida de usuario del sistema			Fuente	David Tedejo García		
Descripción	El sistema controlará la salida del usuario del sistema, de manera que no se quede la sesión abierta en ningún equipo.						
Prioridad	Alta	Media	Baja	Necesidad	Alta	Media	Baja
Estabilidad	Alta	Media	Baja	Riesgo	Alta	Media	Baja
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 28: Requisito RSEG-002

ID	RSEG-003			Categoría	Validación		
Nombre	Validación de campos en formularios			Fuente	David Tedejo García		
Descripción	El sistema dispondrá de un sistema de validación en los campos de un formulario, de manera que se comprueban campos numéricos, correo electrónico, fechas, etc.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 29: Requisito RSEG-003

ID	RSEG-004			Categoría	Validación		
Nombre	Comprobación de registro			Fuente	David Tedejo García		
Descripción	El sistema dispondrá de una comprobación de los registros de usuario de manera que no pueda cometerse ningún error en el mismo, nombres inválidos o nombres ya existente.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 30: Requisito RSEG-004

ID	RSEG-005			Categoría		Validación	
Nombre	Comprobación campos obligatorios			Fuente		David Tedejo García	
Descripción	El sistema dispondrá de una comprobación de los campos obligatorios de manera que no se puedan dejar en blanco aquellos que hayan sido marcados como obligatorios.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 31: Requisito RSEG-005

6.3.7. Requisitos de interfaz

ID	RINT-001			Categoría	Interfaz		
Nombre	Panel de control de usuario			Fuente	David Tedejo García		
Descripción	La interfaz dispondrá de un panel de control para el usuario de manera que pueda navegar fácilmente por la aplicación.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 32: Requisito RINT-001

ID	RINT-002			Categoría	Interfaz		
Nombre	Formulario registro			Fuente	David Tedejo García		
Descripción	La interfaz dispondrá de un formulario de registro de usuario con los campos necesarios para ello.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 33: Requisito RINT-002

ID	RINT-003			Categoría	Interfaz		
Nombre	Formulario acceso			Fuente	David Tedejo García		
Descripción	La interfaz dispondrá de un formulario para permitir el acceso a la aplicación del usuario, el cual contendrá los campos email y contraseña.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 34: Requisito RINT-003

ID	RINT-004			Categoría	Interfaz		
Nombre	Formulario modificación perfil			Fuente	David Tedejo García		
Descripción	La interfaz dispondrá de un formulario de modificación del perfil de usuario, de manera que se le proporcione al usuario la opción de realizar cualquier cambio sobre su información.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 35: Requisito RINT-004

ID	RINT-005			Categoría	Interfaz		
Nombre	Formulario creación de viajes			Fuente	David Tedejo García		
Descripción	La interfaz dispondrá de un formulario para la creación de viajes que vaya a realizar cada usuario.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 36: Requisito RINT-005

ID	RINT-006			Categoría	Interfaz		
Nombre	Formulario modificación de viajes			Fuente	David Tedejo García		
Descripción	La interfaz dispondrá de un formulario para la modificación de viajes antes de que el usuario los realice.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 37: Requisito RINT-006

ID	RINT-007			Categoría		Interfaz	
Nombre	Información para contacto			Fuente		David Tedejo García	
Descripción	Se dispondrá de una interfaz donde el usuario pueda ver la información para contactar con ShareCar .						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 38: Requisito RINT-007

ID	RINT-008			Categoría	Interfaz		
Nombre	Soporte de navegadores			Fuente	David Tedejo García		
Descripción	La interfaz dispondrá de soporte para que la aplicación pueda ser utilizada en distintos navegadores.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 39: Requisito RINT-008

6.3.8. Requisitos de rendimiento

ID	RREN-001			Categoría		Rendimiento	
Nombre	Tiempo de carga interfaz			Fuente		David Tedejo García	
Descripción	En el sistema se tendrá en cuenta el tiempo que tarda la interfaz en cargar en los diferentes navegadores de manera que no supere tres segundos de carga.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 40: Requisito RREN-001

ID	RREN-002			Categoría		Rendimiento	
Nombre	Cierre de sesión			Fuente		David Tedejo García	
Descripción	El cierre de sesión no podrá superar tres segundos.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 41: Requisito RREN-002

ID	RREN-003			Categoría		Rendimiento	
Nombre	Tiempo inicio sesión			Fuente		David Tedejo García	
Descripción	El inicio de sesión no podrá superar tres segundos.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 42: Requisito RREN-003

6.4. Especificación de casos de uso

En esta sección se van a detallar todos los casos de uso del sistema, un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo un proceso. Por lo tanto, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores (personajes o entidades que participan en el caso de uso).

A continuación, se pueden observar las distintas interacciones entre los actores y el sistema ofrecido:

1. Actor:

- Usuario administrador.

• Interacciones con el sistema:

- Ver estado servidores.
- Denegación de acceso.
- Aplicar cambios en el sistema.

2. Actor:

- Usuario no registrado.

• Interacciones con el sistema:

- Registrarse.
- Buscar viajes.
- Ver perfil de usuarios.
- Ver usuarios mejor puntuados.
- Ver información de contacto de **ShareCar**.
- Ver comentarios sobre **ShareCar**.

3. Actor:

- Usuario registrado.

• Interacciones con el sistema:

- Iniciar sesión
 - Buscar viajes.
 - Crear viajes.
 - Modificar viajes propios.
 - Eliminar viajes propios.
 - Ver viajes realizados.
 - Puntuar otros usuarios.
 - Ver usuarios mejor puntuados.

- Comentar sobre otros usuarios.
- Comentar sobre **ShareCar**.
- Modificar perfil.
- Ver puntuación de otro usuario.
- Ver comentarios de otro usuario.
- Ver comentarios sobre **ShareCar**.
- Ver información de contacto de **ShareCar**.
- Ver usuarios recomendados por **ShareCar**.
- Darse de baja.
- Cerrar sesión.

Una vez que se han especificado todas las interacciones entre actores y sus interacciones con el sistema, se va a realizar un estudio y un análisis más completo de cada uno de los casos de uso expuestos.

Primero se van a mostrar diagramas de casos de uso divididos por los diferentes actores que los realizan:



Ilustración 6: Diagrama casos de uso usuario administrador



Ilustración 7: Diagrama casos de uso usuario no registrado

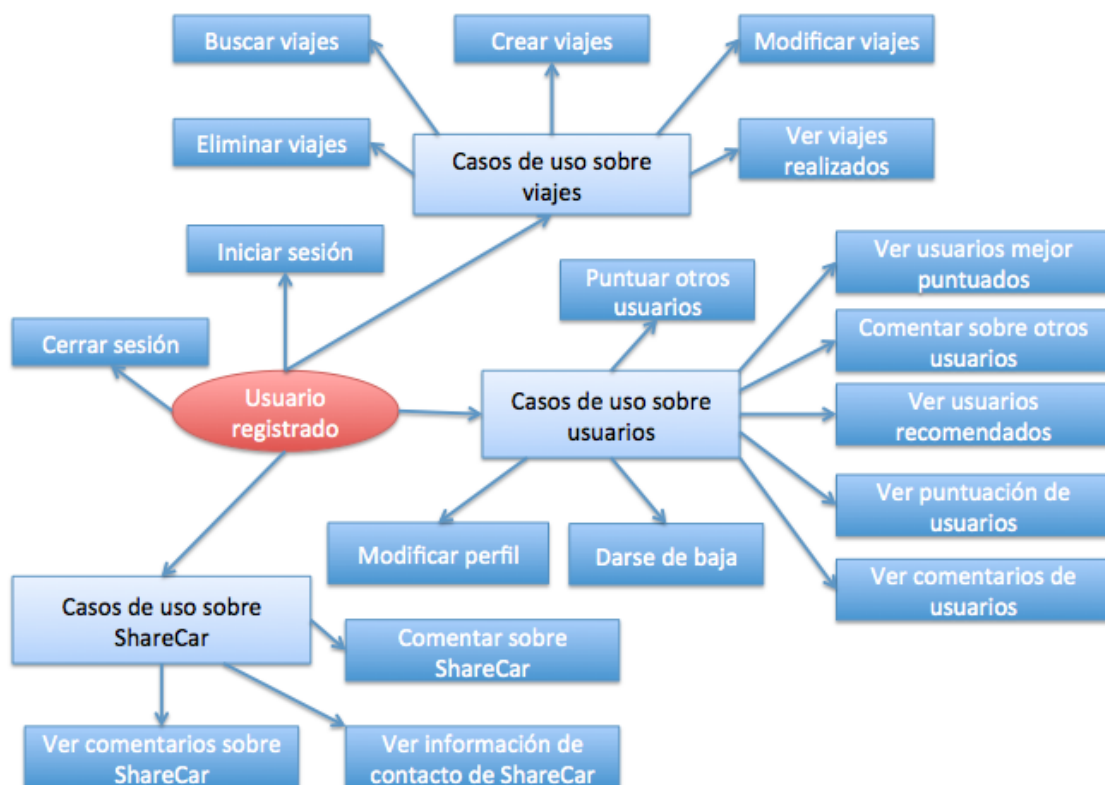


Ilustración 8: Diagrama casos de uso usuario registrado

Para especificar mejor cada uno de los casos de uso anteriores entre los actores y el sistema se va a utilizar la siguiente plantilla de tablas:

CASO DE USO	
Id (CU-XXX)	Nombre del caso de uso
Actor/Actores	Actor/es que participan en el caso de uso
Objetivo	Función o fin del caso de uso
Pre-condiciones	Aquellas acciones que se deben cumplir para poder realizar el caso de uso
Post-condiciones	Aquellas situaciones que aparecen cuando se ha realizado el caso de uso

Tabla 43: Sistema de tablas para los casos de uso

Una vez especificado el sistema de tablas que se va a utilizar para la especificación detallada de los casos de uso, se empieza el análisis completo de los casos de uso del sistema.

CASO DE USO	
CU-001	Ver estado de servidores
Actor/Actores	Administrador
Objetivo	El administrador del sistema puede conocer en todo momento el estado de los servidores del mismo.
Pre-condiciones	El administrador debe conocer el funcionamiento y organización de los servidores.
Post-condiciones	El administrador visualizará el estado de los servidores.

Tabla 44: Caso de uso CU-001

CASO DE USO	
CU-002	Denegación de acceso
Actor/Actores	Administrador
Objetivo	El administrador del sistema puede denegar el acceso a los usuarios por motivos de seguridad o de indisciplina.
Pre-condiciones	El administrador debe llevar un control sobre los usuarios del sistema.
Post-condiciones	El usuario queda bloqueado y se le deniega la entrada en el sistema.

Tabla 45: Caso de uso CU-002

CASO DE USO	
CU-003	Aplicar cambios sobre el sistema
Actor/Actores	Administrador
Objetivo	El administrador del sistema puede aplicar cambios en el mismo cuando lo estime oportuno y sea para mejorar el mismo.
Pre-condiciones	El administrador debe conocer el funcionamiento y la organización del sistema.
Post-condiciones	Los cambios en el sistema se llevan a cabo.

Tabla 46: Caso de uso CU-003

CASO DE USO	
CU-004	Registrarse
Actor/Actores	Usuario no registrado
Objetivo	Un usuario que no tenga cuenta puede crear una mediante el formulario de registro.
Pre-condiciones	No hay.
Post-condiciones	El usuario queda registrado en el sistema.

Tabla 47: Caso de uso CU-004

CASO DE USO	
CU-005	Buscar viajes
Actor/Actores	Usuario no registrado y usuario registrado
Objetivo	Cualquier usuario puede buscar viajes que concuerden con sus necesidades.
Pre-condiciones	No hay.
Post-condiciones	Se muestran los viajes que coinciden con sus necesidades.

Tabla 48: Caso de uso CU-005

CASO DE USO	
CU-006	Ver perfil de otros usuarios
Actor/Actores	Usuario no registrado
Objetivo	Cualquier usuario puede acceder al perfil de otro, pero no podrá ver información de contacto ni valorar a dicho usuario.
Pre-condiciones	No hay.
Post-condiciones	El usuario ve algunos datos de otro usuario.

Tabla 49: Caso de uso CU-006

CASO DE USO	
CU-007	Ver usuarios mejor puntuados
Actor/Actores	Usuario no registrado y usuario registrado
Objetivo	Cualquier usuario puede ver los usuarios que tienen mejor puntuación.
Pre-condiciones	No hay.
Post-condiciones	Se muestran los usuarios que tienen mejor puntuación.

Tabla 50: Caso de uso CU-007

CASO DE USO	
CU-008	Ver información de contacto de ShareCar
Actor/Actores	Usuario no registrado y usuario registrado
Objetivo	Cualquier usuario puede ver la información de contacto de ShareCar .
Pre-condiciones	No hay.
Post-condiciones	Se muestra la información de contacto de ShareCar .

Tabla 51: Caso de uso CU-008

CASO DE USO	
CU-009	Ver comentarios sobre ShareCar
Actor/Actores	Usuario no registrado y usuario registrado
Objetivo	Un usuario puede ver los comentarios realizados sobre ShareCar .
Pre-condiciones	El usuario puede ver los comentarios en la información de contacto de ShareCar .
Post-condiciones	El usuario registrado ve los comentarios realizados sobre ShareCar .

Tabla 52: Caso de uso CU-009

CASO DE USO	
CU-010	Iniciar sesión
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede iniciar sesión con su cuenta en el sistema.
Pre-condiciones	El usuario debe haberse registrado con anterioridad.
Post-condiciones	El usuario entra al sistema con su cuenta.

Tabla 53: Caso de uso CU-010

CASO DE USO	
CU-011	Crear viajes
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede crear viajes que vaya a realizar.
Pre-condiciones	El usuario debe haber iniciado sesión.
Post-condiciones	El usuario crea un nuevo viaje.

Tabla 54: Caso de uso CU-011

CASO DE USO	
CU-012	Modificar viajes
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede modificar los viajes que ha creado anteriormente.
Pre-condiciones	El usuario debe haber iniciado sesión y haber creado un viaje.
Post-condiciones	El usuario modifica el viaje.

Tabla 55: Caso de uso CU-012

CASO DE USO	
CU-013	Eliminar viajes
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede eliminar (cancelar) viajes antes de realizarlos.
Pre-condiciones	El usuario debe haber iniciado sesión y haber creado un viaje.
Post-condiciones	El usuario elimina el viaje.

Tabla 56: Caso de uso CU-013

CASO DE USO	
CU-014	Ver viajes realizados
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede ver los viajes que ha realizado.
Pre-condiciones	El usuario debe haber iniciado sesión y haber realizado algún viaje.
Post-condiciones	El usuario ve los viaje realizados.

Tabla 57: Caso de uso CU-014

CASO DE USO	
CU-015	Puntuar a otros usuarios
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede puntuar a otros usuarios con valores comprendidos entre uno y cinco.
Pre-condiciones	El usuario debe haber iniciado sesión y acceder al perfil del otro usuario.
Post-condiciones	El usuario registrado puntúa al otro usuario.

Tabla 58: Caso de uso CU-015

CASO DE USO	
CU-016	Comentar sobre otros usuarios
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede comentar sobre otros usuarios.
Pre-condiciones	El usuario debe haber iniciado sesión y acceder al perfil del otro usuario.
Post-condiciones	El usuario registrado realiza un comentario sobre el otro usuario.

Tabla 59: Caso de uso CU-016

CASO DE USO	
CU-017	Comentar sobre ShareCar
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede comentar sobre ShareCar .
Pre-condiciones	El usuario debe haber iniciado sesión y acceder a su perfil de usuario.
Post-condiciones	El usuario registrado realiza un comentario sobre ShareCar .

Tabla 60: Caso de uso CU-017

CASO DE USO	
CU-018	Modificar perfil
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede modificar los datos de su perfil y de su cuenta.
Pre-condiciones	El usuario debe haber iniciado sesión y acceder al formulario de modificación de perfil.
Post-condiciones	El usuario modifica los datos que le interesan de su perfil.

Tabla 61: Caso de uso CU-018

CASO DE USO	
CU-019	Ver puntuación de otro usuario
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede ver la puntuación de otro usuario en el perfil de este último.
Pre-condiciones	El usuario debe haber iniciado sesión y acceder al perfil del otro usuario.
Post-condiciones	El usuario registrado ve la puntuación media del otro usuario.

Tabla 62: Caso de uso CU-019

CASO DE USO	
CU-020	Ver comentarios de otro usuario
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede ver los comentarios realizados sobre otro usuario en el perfil de este último.
Pre-condiciones	El usuario debe haber iniciado sesión y acceder al perfil del otro usuario.
Post-condiciones	El usuario registrado ve los comentarios realizados sobre el otro usuario.

Tabla 63: Caso de uso CU-020

CASO DE USO	
CU-021	Ver usuarios recomendados por ShareCar
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede ver los usuarios con perfiles similares al suyo recomendados por ShareCar .
Pre-condiciones	El usuario debe haber iniciado sesión y acceder a su perfil.
Post-condiciones	El usuario registrado ve los usuarios recomendados por ShareCar .

Tabla 64: Caso de uso CU-021

CASO DE USO	
CU-022	Darse de baja
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede darse de baja cuando desee.
Pre-condiciones	El usuario debe haber iniciado sesión y acceder a su perfil.
Post-condiciones	El usuario da de baja su cuenta y se borra su perfil de la aplicación.

Tabla 65: Caso de uso CU-022

CASO DE USO	
CU-023	Cerrar sesión
Actor/Actores	Usuario registrado
Objetivo	Un usuario registrado puede cerrar sesión cuando desee.
Pre-condiciones	El usuario debe haber iniciado sesión.
Post-condiciones	El usuario cierra su sesión en la aplicación.

Tabla 66: Caso de uso CU-023

Por último, se muestra la matriz de trazabilidad entre los requisitos expuesto en 6.3 Especificación de requisitos software y los casos de uso que se acaban de mostrar:

Ids.	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015	CU-016	CU-017	CU-018	CU-019	CU-020	CU-021	CU-022	CU-023
RUSU-001				X																			
RUSU-002																						X	
RUSU-003																		X					
RUSU-004										X													
RUSU-005																							X
RUSU-006						X																	
RUSU-007																		X					
RUSU-008						X																	
RVIA-001				X																			
RVIA-002										X													
RVIA-003												X											
RVIA-004											X												
RVIA-005													X										

Ids.	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015	CU-016	CU-017	CU-018	CU-019	CU-020	CU-021	CU-022	CU-023
RVIA-006					X																		
RPUN-001						X													X				
RPUN-002															X								
RPUN-003							X																
RCOM-001						X														X			
RCOM-002								X															
RCOM-003																X							
RCOM-004																	X						
RADM-001		X																					
RADM-002	X																						
RADM-003			X																				
RSEG-001						X																	
RSEG-002																							X
RSEG-003				X							X												
RSEG-004				X																			
RSEG-005				X							X												
RINT-001				X	X	X		X		X	X	X	X	X	X	X	X	X				X	X
RINT-002				X																			
RINT-003										X													
RINT-004																		X					
RINT-005											X												
RINT-006												X											
RINT-007								X															

Ids.	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015	CU-016	CU-017	CU-018	CU-019	CU-020	CU-021	CU-022	CU-023
RINT-008				X	X	X		X		X	X	X	X	X	X	X	X	X				X	X
RREN-001				X	X	X		X			X	X	X	X	X	X	X	X				X	
RREN-002																							X
RREN-003										X													

Tabla 67: Trazabilidad Requisitos software-Casos de uso

7. Diseño del sistema

En esta sección se define el sistema buscando dar una visión mucho más concreta de lo que será el software y supondrá un enfoque mayor de todas las características con las que contará el sistema.

Para ello, se realiza una definición de la arquitectura del sistema, especificando niveles de arquitectura. Posteriormente se definen los requisitos de diseño específicos para el sistema y el modelo de clases de diseño. Y por último, se especifica el diseño físico de datos, es decir, la estructura que tendrá la base de datos del sistema.

7.1. Definición de la arquitectura del sistema

Ahora se define a alto nivel las estructuras de agrupamiento de los distintos componentes del sistema. Cada agrupación contendrá las partes significativas del sistema que tienen características propias de función o ejecución.

Para la realización de esta agrupación se utiliza el Modelo-Vista-Controlador (MVC) dado que se ha considerado que es la más eficiente para un sistema como este en el que el usuario accederá a través de un navegador web.

En ella existen tres niveles de arquitectura:

- **Vista:** agrupará todas aquellas clases encargadas de presentar el 'Modelo' en un formato adecuado para interactuar con el usuario.
- **Modelo:** es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones. Envía a la 'Vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada. Las peticiones de acceso o manipulación de información llegan al 'Modelo' a través del 'Controlador'.
- **Controlador:** agrupará todas aquellas clases encargadas de responder a eventos del usuario e invocar peticiones al 'Modelo' cuando se necesita información del sistema. También puede enviar comandos a la 'Vista' para cambiar la forma en que se presenta el 'Modelo'. Se podría decir que hace de intermediario entre el 'Modelo' y la 'Vista'.

7.2. Requisitos de diseño

A continuación se especifican los requisitos que se definen para cubrir las necesidades de diseño y construcción del sistema. Mediante la definición de los siguientes requisitos se espera cumplir las funcionalidades requeridas para el correcto funcionamiento de **ShareCar**.

Las siguientes tablas siguen el modelo usado para la especificación de los requisitos creados con anterioridad en este documento. Se especificarán los requisitos extraídos para el diseño y la construcción del sistema.

ID	RDC-001			Categoría	Diseño y Construcción		
Nombre	Arquitectura del sistema			Fuente	David Tedejo García		
Descripción	La arquitectura usada para la elaboración del sistema se basa en el Modelo-Vista-Controlador.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 68: Requisito RDC-001

ID	RDC-002			Categoría	Diseño y Construcción		
Nombre	Almacenamiento físico			Fuente	David Tedejo García		
Descripción	Cada uno de los servidores debe disponer de una capacidad de almacenamiento físico necesariamente alta para almacenar la información del sistema.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 69: Requisito RDC-002

ID	RDC-003			Categoría	Diseño y Construcción		
Nombre	Velocidad de conexión			Fuente	David Tedejo García		
Descripción	Se tendrá una conexión a Internet lo suficientemente rápida para cumplir los requisitos expuestos en 6.3 Especificación de requisitos software.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 70: Requisito RDC-003

ID	RDC-004			Categoría	Diseño y Construcción		
Nombre	Lenguaje de programación			Fuente	David Tedejo García		
Descripción	El lenguaje de programación utilizado para el desarrollo del sistema es el establecido con anterioridad en la sección 5.3.2 Software.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 71: Requisito RDC-004

ID	RDC-005			Categoría	Diseño y Construcción		
Nombre	Servidores			Fuente	David Tedejo García		
Descripción	Los servidores utilizados en el desarrollo de la aplicación serán Apache HTTP <i>Server</i> .						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 72: Requisito RDC-005

ID	RDC-006			Categoría	Diseño y Construcción		
Nombre	Almacenamiento en BBDD			Fuente	David Tedejo García		
Descripción	El sistema utilizado para el almacenamiento de la base de datos será <i>MySQL Community Edition</i> .						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 73: Requisito RDC-006

ID	RDC-007			Categoría	Diseño y Construcción		
Nombre	Diseño web adaptable (<i>responsive web design</i>)			Fuente	David Tedejo García		
Descripción	Se utilizará la filosofía de diseño <i>responsive</i> para conseguir adaptar la aplicación web al entorno del usuario.						
Prioridad	Alta	Media	Baja	Necesidad	Necesario	Deseable	Opcional
Estabilidad	Alta	Media	Baja	Riesgo	Alto	Moderado	Bajo
Verificabilidad	Alta	Media	Baja	Claridad	Alta	Media	Baja

Tabla 74: Requisito RDC-007

7.3. Diseño de clases

A continuación se va a elaborar el **modelo de clases de diseño**. Para ello, se tendrán en cuenta los requisitos del análisis del sistema, así como, aspectos tecnológicos que podrán hacer variar las clases.

7.3.1. Identificación de clases de diseño

A continuación se identificarán las diferentes clases del sistema. Se empieza con un primer diseño más global del sistema, para ofrecer una visión global del mismo y se irá profundizando posteriormente.

Primero se muestra el diagrama de interacción entre los diferentes componentes y clases correspondientes a la aplicación. Las clases de los cuadrados pertenecen al subsistema vista, y las clases del subsistema controlador se representan como interacciones entre las anteriores mediante flechas rojas numeradas:

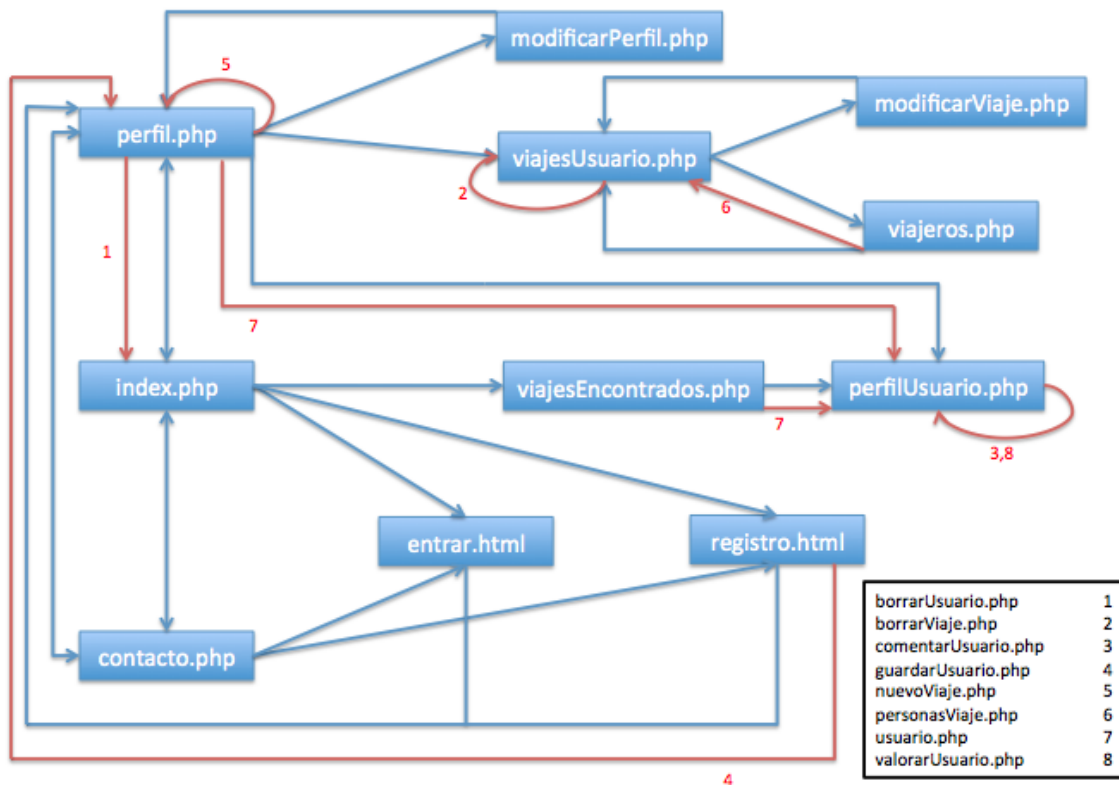


Ilustración 9: Diagrama de interacciones entre componentes

En este diagrama faltaría indicar que desde cualquier clase del subsistema vista se puede acceder a las de *index.php*, *perfil.php* y *contacto.php*. También se debe indicar que faltarían del subsistema controlador la clase *cerrarSesion.php*, a la que se puede acceder desde todas una vez se haya iniciado sesión, y la clase *login.php*, a la que se puede acceder desde *index.php*, *entrar.html*, *contacto.php*, *viajesEncontrados.php* y *perfilUsuario.php*. Estas características no se han incluido en el diagrama a fin de mejorar la legibilidad del mismo, no sobrecargándolo.

Al usar una arquitectura Modelo-Vista-Controlador las clases se han dividido en las que están destinada a la Vista, las que están destinadas al Controlador y la parte destinada al Modelo.

- **Subsistema Vista:** son las clases destinadas a ofrecer la parte visual al usuario y éstas son las siguientes:
 - *contacto.php*, es la página de contacto de **ShareCar**.
 - *entrar.html*, página mediante la cual se puede iniciar sesión, es a donde se redirige al usuario si éste intenta acceder a su perfil y no ha iniciado sesión previamente.
 - *index.php*, es la página inicial de la aplicación **ShareCar**.
 - *modificarPerfil.php*, es la página mediante la cual se pueden modificar los datos del perfil personal de cada usuario.
 - *modificarViaje.php*, página en la que el usuario puede modificar datos de un viaje que ha creado con anterioridad y que todavía no ha realizado.
 - *perfil.php*, es la página destinada a mostrar la información del usuario que ha iniciado sesión en el sistema.
 - *perfilUsuario.php*, página destinada a mostrar el perfil de un usuario que no sea el que está interactuando con el sistema en ese momento.

- *registro.html*, es la página a través de la cual se realiza el registro de un usuario.
- *viajeros.php*, página en la cual el usuario conductor de un viaje indica quiénes son los demás usuarios que han viajado con él.
- *viajesEncontrados.php*, es la página destinada a mostrar los viajes que se han encontrado cuando el usuario realiza una búsqueda de éstos.
- *viajesUsuario.php*, página mediante la cual el usuario puede visualizar los viajes que ha realizado o que tiene pendientes para realizar en un futuro.
- *Clases CSS*, son clases de estilo que sirven para describir el aspecto y el formato de las clases anteriores.
- **Subsistema Modelo:** en esta parte están los objetos que representan los datos del sistema, es decir, todos los datos relativos a la información de los usuarios, sus viajes y todo lo relacionado con ambos. Todo ello se guarda en una base de datos para poder interactuar de una manera sencilla con los datos.
- **Subsistema Controlador:** aquí se incluyen las clases destinadas a responder a eventos, que llegan del 'Subsistema Vista', e invocar peticiones al 'Subsistema Modelo'. Se puede decir que actúa de intermediario entre los dos, y dichas clases son:
 - *borrarUsuario.php*, clase destinada a borrar a un usuario del sistema cuando éste quiera darse de baja.
 - *borrarViaje.php*, clase destinada a borrar un viaje del sistema cuando el usuario ha decidido hacerlo.
 - *cerrarSesion.php*, clase mediante la cual se cierra la sesión del usuario.
 - *comentarUsuario.php*, clase que se encarga de guardar los comentarios sobre otros usuarios o sobre **ShareCar** de un usuario que ha iniciado sesión previamente.
 - *guardarUsuario.php*, clase que se encarga de llevar a cabo el registro de un usuario en el sistema y guardar toda su información.
 - *login.php*, clase encargada del inicio de sesión de un usuario en la aplicación.
 - *nuevoViaje.php*, clase destinada a la creación de nuevos viajes con los datos introducidos por el usuario.
 - *personasViajes.php*, clase mediante la cual se relaciona los viajes y su conductor con las personas que han viajado de acompañantes.
 - *usuario.php*, clase destinada a diferenciar los perfiles de otros usuarios que se quieren visualizar.
 - *valorarUsuario.php*, clase que se encarga de guardar las puntuaciones sobre otros usuarios de un usuario que ha iniciado sesión previamente.
 - *Clases JavaScript*, clase que sirve para capturar eventos y generar acciones.

7.4. Diseño físico de datos

En esta actividad se define la estructura física de datos que utilizará el sistema a partir del modelo de clases, de manera que teniendo presentes las características específicas del sistema de gestión de datos concreto a utilizar, los requisitos específicos para el sistema de información, y las particularidades del entorno tecnológico se consiga una mayor eficiencia en el tratamiento de los datos.

7.4.1. Diseño del modelo físico de datos

El objetivo de esta tarea es realizar el diseño del modelo físico de datos a partir del modelo de clases y de los requisitos del sistema.

Destacar que el gestor de base de datos utilizado es “MySQL Community Edition” como se especificó anteriormente en el presente documento.

A continuación se muestra el modelo físico de base de datos:

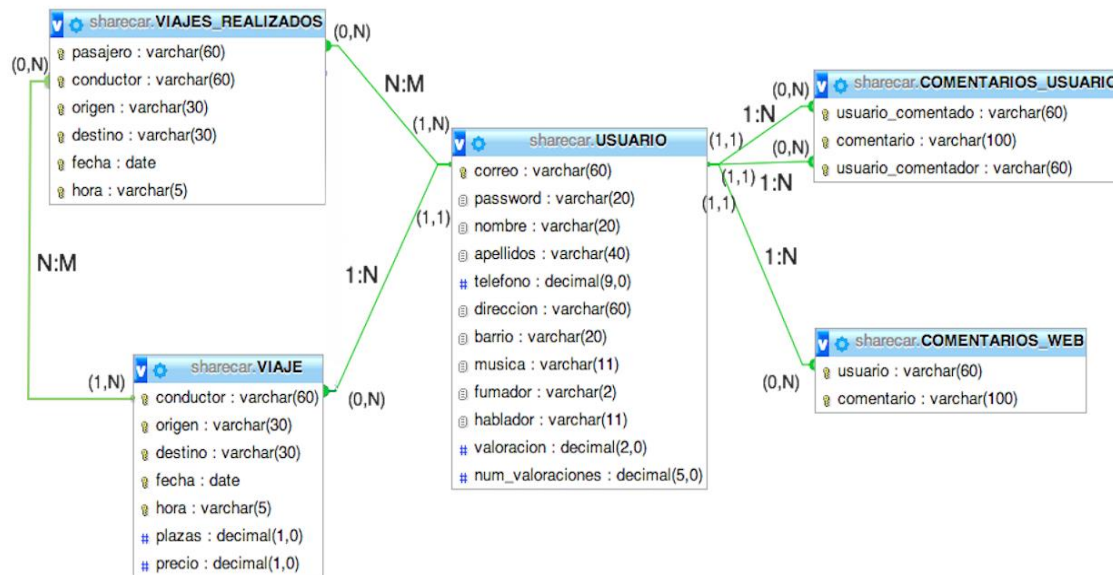


Ilustración 10: Modelo físico de datos

Seguidamente se pasa a describir cada una de las tablas establecidas en el modelo físico de datos, así como los atributos correspondientes a cada una, además de especificar las claves primarias (PK) y las claves alternativas (FK).

Las distintas tablas creadas en el modelo físico de datos son:

Usuario		
Descripción	Esta tabla contiene todos los datos de cada uno de los usuarios registrados en la aplicación.	
Atributos	<ul style="list-style-type: none"> - correo: VARCHAR(60) NOT NULL - password: VARCHAR(20) NOT NULL - nombre: VARCHAR(20) NOT NULL - apellidos: VARCHAR(40) NOT NULL - telefono: DECIMAL(9,0) NOT NULL - direccion: VARCHAR(60) NOT NULL - barrio: VARCHAR(20) NOT NULL - musica: VARCHAR(11) NOT NULL - fumador: VARCHAR(2) NOT NULL - hablador: VARCHAR(11) NOT NULL - valoracion: DECIMAL(2,0) - num_valoraciones: DECIMAL(5,0) - PK: correo 	

Tabla 75: Tabla usuario

Viaje	
Descripción	Esta tabla almacena todos los datos relacionados con los viajes creados por cada usuario.
Atributos	<ul style="list-style-type: none"> - conductor: VARCHAR(60) NOT NULL - origen: VARCHAR(30) NOT NULL - destino: VARCHAR(30) NOT NULL - fecha: DATE NOT NULL - hora: VARCHAR(5) NOT NULL - plazas: DECIMAL(1,0) NOT NULL - precio: DECIMAL(1,0) NOT NULL - PK: conductor, origen, destino, fecha, hora - FK(usuario): conductor

Tabla 76: Tabla viaje

Viajes_realizados	
Descripción	Esta tabla intermedia almacena la relación entre los viajes creados por un usuario y los pasajeros de dicho viaje.
Atributos	<ul style="list-style-type: none"> - pasajero: VARCHAR(60) NOT NULL - conductor: VARCHAR(60) NOT NULL - origen: VARCHAR(30) NOT NULL - destino: VARCHAR(30) NOT NULL - fecha: DATE NOT NULL - hora: VARCHAR(5) NOT NULL - plazas: DECIMAL(1,0) NOT NULL - precio: DECIMAL(1,0) NOT NULL - PK: pasajero, conductor, origen, destino, fecha, hora - FK(usuario): pasajero - FK(viaje): conductor, origen, destino, fecha, hora

Tabla 77: Tabla viajes_realizados

Comentarios_usuario	
Descripción	Esta es una tabla que contiene la información relacionada entre los usuarios y los comentarios que hacen sobre otros usuarios.
Atributos	<ul style="list-style-type: none"> - usuario_comentado: VARCHAR(60) NOT NULL - comentario: VARCHAR(100) NOT NULL - usuario_comentador: VARCHAR(60) NOT NULL - PK: usuario_comentado, comentario, usuario_comentador - FK(usuario): usuario_comentado - FK(usuario): usuario_comentador

Tabla 78: Tabla comentarios_usuario

Comentarios_web	
Descripción	Esta tabla almacena los datos correspondientes a los comentarios que realizan los usuarios sobre la página web ShareCar .
Atributos	<ul style="list-style-type: none"> - usuario: VARCHAR(60) NOT NULL - comentario: VARCHAR(100) NOT NULL - PK: usuario, comentario - FK(usuario): usuario

Tabla 79: Tabla comentarios_web

7.4.2. Especificación de los caminos de acceso a los datos y optimización

El objetivo de esta tarea es determinar los caminos de acceso a los datos persistentes en el sistema, utilizados por las principales clases, con el fin de optimizar el rendimiento y consumo de recursos, así como disminuir los tiempos de respuesta.

Gracias al modelo físico de datos especificado en el apartado anterior, se pueden observar los distintos caminos de acceso a los datos necesarios para mostrárselos al usuario.

De momento, dado que es una base de datos sencilla, no se necesita realizar ningún tipo de indización para mejorar el tiempo de búsqueda de resultados. Si con el tiempo el número de usuarios fuera extremadamente alto podría haber problemas de escalabilidad, entonces sí que se necesitarían índices, por ejemplo, sobre las claves o, realizando un estudio, sobre elementos que se utilicen mucho.

En el peor de los casos, a la hora de acceder a un dato, será necesario acceder a una tabla intermedia en el caso de que la relación sea M:N.

En los demás casos, sólo es necesario acceder a una tabla para obtener los resultados necesarios para posteriormente mostrárselos a los usuarios.

7.4.3. Especificación de la distribución de datos

En esta sección se determina el modelo de distribución de datos, teniendo en cuenta la arquitectura del sistema que se ha especificado. Se establece la ubicación de los gestores de bases de datos, sistemas de ficheros y el resto de elementos de la estructura física de datos en los nodos correspondientes de los que se han identificado en la arquitectura.

En nuestro caso se va a almacenar toda la información en un servidor principal, del cual se realizarán copias de seguridad de forma continua y diaria en otro servidor secundario redundante, para evitar la pérdida de información en caso de que el servidor principal se dañe y con el fin de evitar cualquier conflicto con los distintos usuarios del sistema.

8. Implementación del prototipo

En esta sección se explica el prototipo que se ha implementado para conseguir una primera versión o versión beta de la aplicación **ShareCar**. Para ello se explica la estructura y organización de todas las clases y ficheros necesarios para la implementación de dicho prototipo y las interfaces resultantes tras la implementación.

Por otro lado, cabe destacar que el citado prototipo ha sido implementado y se incluye en un archivo adjunto a esta memoria llamado 'ShareCar.zip', el cual contiene, por un lado, la estructura que se explica en la siguiente sección perteneciente al prototipo y sus interfaces y, por otro lado, los ficheros necesarios para crear las tablas e insertar algunos datos en la base de datos necesaria para la ejecución de la aplicación.

8.1. Estructura del prototipo

A continuación se va a exponer la estructura que tiene el prototipo desarrollado para este proyecto. Destacar que se ha usado una biblioteca llamada '*Bootstrap*' para agilizar la implementación del proyecto y para utilizar los recursos que ofrece esta biblioteca para hacer páginas *web responsive*, la cual se basa en una filosofía de diseño y desarrollo web que mediante el uso de estructuras e imágenes fluidas consigue adaptar el sitio web al entorno del usuario.

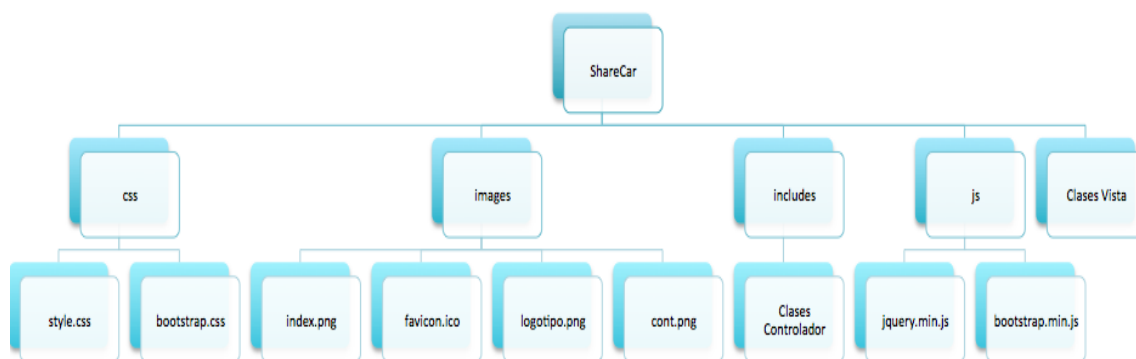


Ilustración 11: Estructura prototipo

El prototipo está dividido en carpetas según la funcionalidad de cada elemento, y ahora se explica más detalladamente:

- **css:** en esta carpeta se guardan todas las hojas de estilo del prototipo, las cuales son:
 - *bootstrap.css*: hoja de estilo de la biblioteca usada con el mismo nombre.
 - *style.css*: hoja de estilo desarrollada durante la implementación del prototipo para ofrecer formatos específicos a **ShareCar**.
- **images:** carpeta en la que se almacenan todas las imágenes que se van a usar en la página web, en ella se encuentran las siguientes imágenes:
 - *cont.png*: imagen usada en la interfaz destinada a contacto.
 - *favicon.ico*: icono usado para mostrar en la barra de direcciones o en la pestaña correspondiente del navegador utilizado.

- *index.png*: imagen elegida para mostrar en la interfaz principal de **ShareCar**.
- *logotipo.png*: logotipo de la página web.
- **includes**: en esta carpeta se guardan todas las clases que implementan el 'Controlador' de la arquitectura elegida Modelo-Vista-Controlador. Dichas clases ya han sido explicadas en la sección 7.3.2 Identificación de clases de diseño.
- **js**: carpeta en la que se guardan las clases destinadas al lenguaje JavaScript y a la biblioteca de éste, JQuery. Se utilizan para permitir mejoras en la interfaz de usuario y estas clases son:
 - *bootstrap.min.js*: clase de la biblioteca utilizada '*Bootstrap*'.
 - *jquery.min.js*: clase que utiliza una biblioteca de JavaScript llamada JQuery. Esta clase y la anterior son la versión reducida (min) para agilizar el proyecto y su carga y así conseguir que éste tenga menos peso.
- **Clases vista**: estas clases son las interfaces que el usuario ve y con las que interactúa, es la 'Vista' de la arquitectura elegida. Dichas clases ya han sido explicadas anteriormente en la sección 7.3.2 Identificación de clases de diseño.

8.2. Diseño de las interfaces del prototipo

Ahora se mostrarán y se explicarán las diferentes interfaces que se han creado y con las que se podrá encontrar cualquier usuario durante el uso de la aplicación.

La primera interfaz que se encuentra un usuario al acceder a la aplicación es la interfaz de inicio:



The screenshot shows the 'ShareCar' application interface. At the top, there is a navigation bar with 'ShareCar', 'Perfil', and 'Contacto' on the left, and a login section on the right with 'Email' and 'Password' input fields, and 'Entrar' and 'Registrarse' buttons. Below the navigation bar is a large 'ShareCar' logo. Underneath the logo is a search section with 'Origen' and 'Destino' dropdown menus (both set to 'Centro'), a 'Fecha (XX/XX/XXXX)' input field (set to 'dd/mm/aaaa'), and 'Buscar »' and 'Limpiar' buttons. At the bottom, there are three columns: 'Viajes disponibles' listing two trips by David Tedejo and Sergio Garcia; 'Usuarios más votados' listing four users with their average ratings; and 'Información' with a description of the service and a button to 'Ir a información y contacto »'.

Ilustración 12: Interfaz inicio

En esta interfaz se puede ver un menú en la parte superior, algo que es fijo en toda la aplicación. Este menú consta de las partes de la aplicación a la izquierda (**ShareCar**, Perfil y Contacto) y a la derecha un formulario para iniciar sesión y un botón para registrarse.

En la parte central de la interfaz se encuentra un formulario con el buscador de viajes, en el cual se pueden introducir el origen, el destino, y la fecha del viaje que se desea encontrar.

Y por último, en la parte inferior, se encuentra información sobre usuarios de la aplicación y sobre la propia aplicación **ShareCar**. Esta parte está dividida en tres, a la izquierda se muestran viajes aleatorios disponibles en este momento, en el centro los usuarios que tienen mejor puntuación media, votada por los demás usuarios y a la derecha información sobre **ShareCar** y un botón para acceder a la interfaz de contacto.

Una interfaz a la que se puede acceder desde el inicio es la de registro de nuevos usuarios:

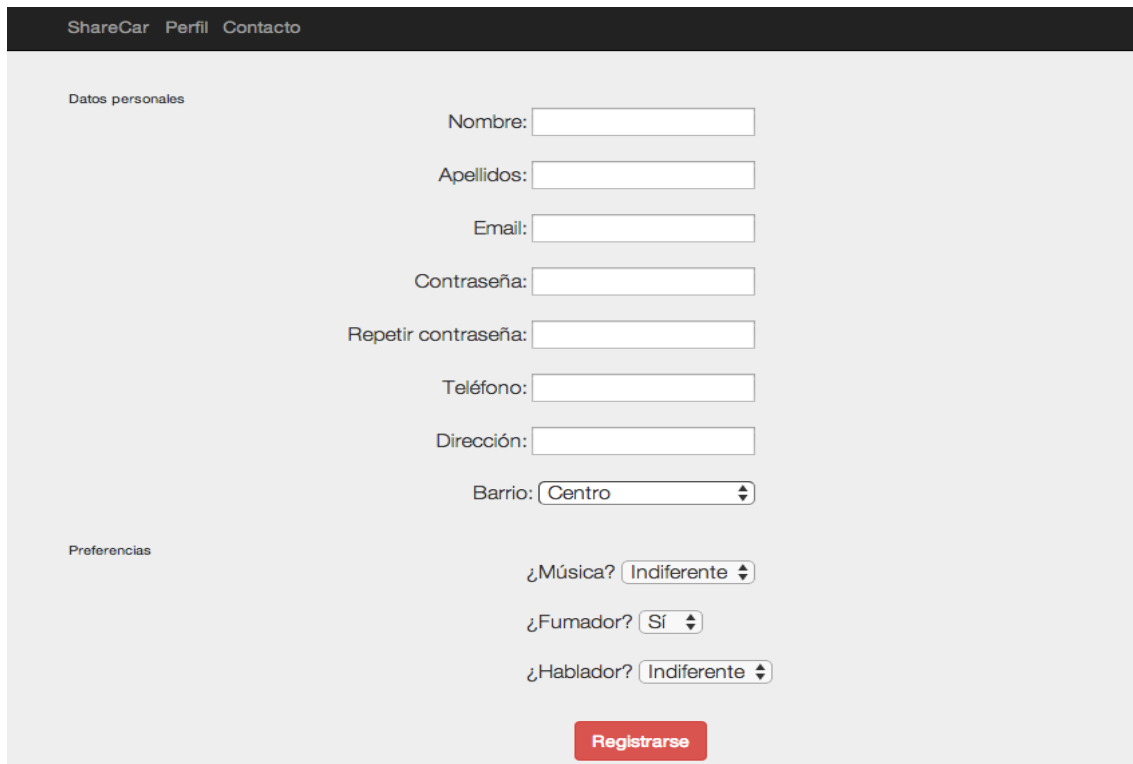


Ilustración 13: Interfaz registro usuarios

Esta interfaz es bastante simple, tiene el menú en la parte superior al igual que todas las demás, pero sin la parte de inicio de sesión ya que se supone que al estar registrándose no podría hacerlo.

También contiene un formulario con todos los campos necesarios para el registro de nuevos usuarios, dichos campos están divididos en datos personales: nombre, apellidos, email, contraseña, repetir contraseña, teléfono, dirección, barrio, y por otra parte las preferencias del usuario: música, fumador, y hablador. Por último está el botón 'Registrarse' para llevar a cabo dicho registro.

Si un usuario quiere acceder a la interfaz perfil sin haber iniciado sesión se le redirige a una interfaz de inicio de sesión:

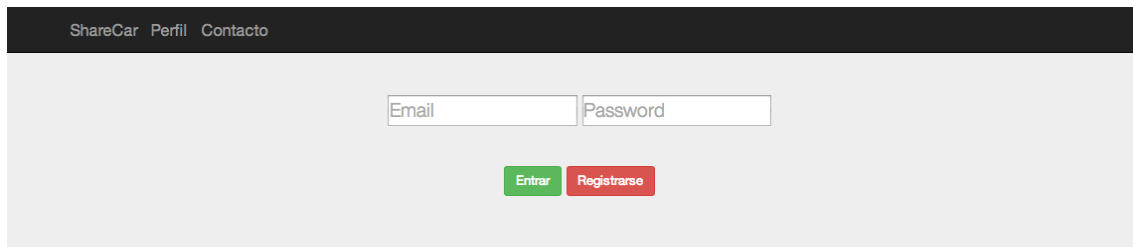


Ilustración 14: Interfaz inicio sesión

La interfaz de inicio de sesión tiene el mismo menú en la parte superior que la interfaz de registro de usuarios y en la parte central un formulario de inicio de sesión, a través del email y la contraseña. También hay un botón de 'Registrarse' para dar lugar a esta opción.

Por otro lado, si un usuario ya ha iniciado sesión y quiere ir al perfil o se acaba de registrar se le redirige a la interfaz de perfil:




Ilustración 15: Interfaz perfil de usuario

Esta interfaz tiene un menú en la parte superior en el que a la izquierda está el menú de navegación de la aplicación y a la izquierda un mensaje de bienvenida y la opción de cerrar sesión y redirigirse a la interfaz de inicio.

En la parte central está el formulario con los campos necesarios para crear un nuevo viaje, del cual el usuario sería el conductor. Para crear un viaje se pide el origen, destino, precio que se va a cobrar a cada acompañante, fecha, hora y número de plazas.

Por último, en la parte inferior se encuentran datos que pueden ser de interés para el usuario como son, de izquierda a derecha: los últimos viajes que ha creado el usuario y la opción de acceder a todos sus viajes, la recomendación de perfiles con la que se le muestran perfiles similares al suyo, la opción de modificar los datos de su perfil y un área de texto para poder poner comentarios sobre **ShareCar**.

Otra interfaz es la destinada a modificar los datos del perfil del usuario:

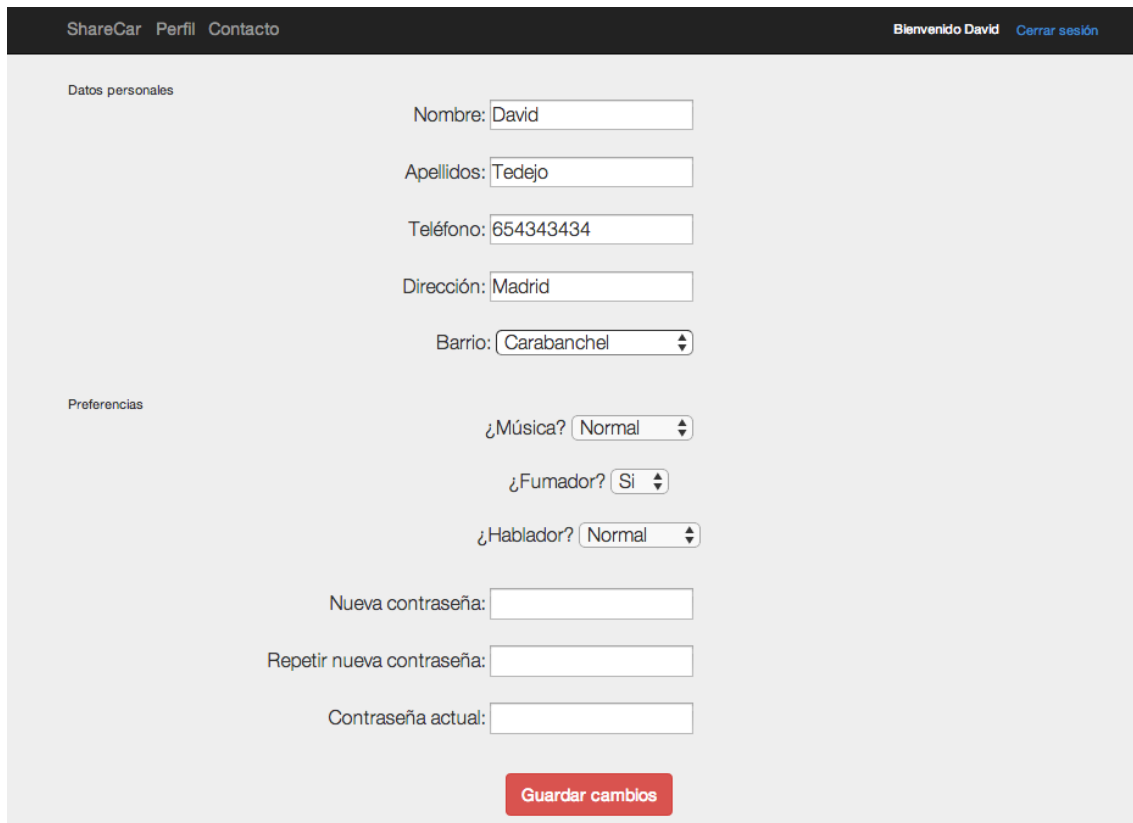


Ilustración 16: Interfaz modificar perfil

Esta interfaz es muy similar la interfaz de registro ya que está el menú en la parte superior con el menú de navegación por la aplicación a la izquierda y el mensaje de bienvenida y la opción de cerrar sesión a la derecha.

En la parte central hay un formulario con los campos que puede cambiar el usuario de su perfil y en los que por defecto vienen los datos del usuario, para poder realizar los cambios el usuario debe introducir su contraseña y si quiere cambiar ésta debe introducirla dos veces para asegurarse de que no hay fallos.

La interfaz que se destinada a mostrar todos los viajes que ha creado el usuario es la siguiente:



Ilustración 17: Interfaz viajes del usuario

Esta interfaz tiene el mismo menú en la parte superior que la interfaz perfil de usuario y en la parte central están todos los viajes que el usuario ha creado, los que ya han sucedido tienen un botón para que el usuario introduzca los acompañantes y los que todavía no han sucedido tienen la opción de modificar los datos del viaje o de cancelar dicho viaje.

Hay una interfaz destinada a indicar los acompañantes de un viaje que ha hecho el usuario como conductor y es la siguiente:

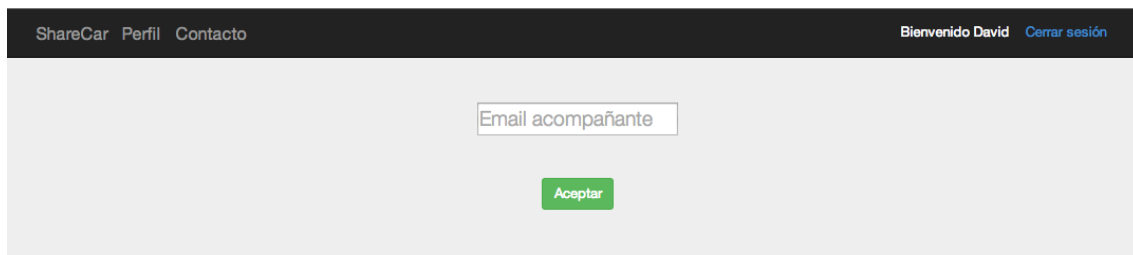


Ilustración 18: Interfaz indicar acompañantes

Esta interfaz es muy simple, en la parte superior tiene un menú que es idéntico al de la interfaz de perfil de usuario y en la parte central un formulario donde hay que indicar el email del usuario que ha ido de acompañante en un viaje.

Otra interfaz relacionada con los viajes es la de modificar los datos de un viaje:



Ilustración 19: Interfaz modificar viaje

Esta interfaz tiene el menú superior igual que en la interfaz perfil de usuario y el parte central un formulario con los campos necesarios para identificar a cada viaje, es el mismo formulario que para crear viajes con la diferencia que en este por defecto vienen los datos del viaje que queremos cambiar.

Se ha creado otra interfaz destinada a mostrar los viajes que se pueden encontrar al realizar una búsqueda de éstos:

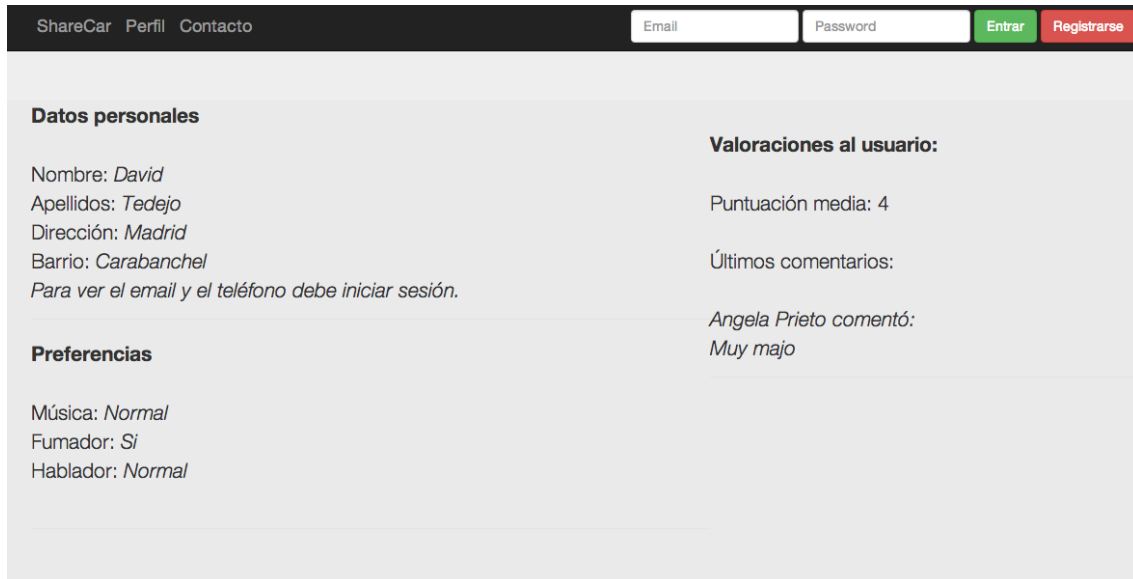


Ilustración 20: Interfaz viajes encontrados

Esta interfaz tiene un menú superior, el cual está dividido en el menú de navegación de la aplicación y a la derecha puede tener el formulario para iniciar sesión o registrarse o el mensaje de bienvenida y la opción de cerrar sesión como tiene el menú superior de la interfaz de perfil de usuario.

En la parte central de la interfaz se muestran todos los viajes que concuerdan con los datos introducidos por el usuario en el formulario de búsqueda de viajes de la interfaz de inicio. Aquí se muestran todos los datos de cada viaje y se puede acceder al perfil del conductor para ponerse en contacto con él.

Hay una interfaz que cambia si el usuario ha iniciado sesión en la aplicación o no, esta interfaz es la de perfil de otro usuario. Primero se muestra si no ha iniciado sesión:



ShareCar Perfil Contacto

Email Password Entrar Registrarse

Datos personales

Nombre: *David*
Apellidos: *Tedejo*
Dirección: *Madrid*
Barrio: *Carabanchel*
Para ver el email y el teléfono debe iniciar sesión.

Preferencias

Música: *Normal*
Fumador: *Si*
Hablador: *Normal*

Valoraciones al usuario:

Puntuación media: 4


Últimos comentarios:

Angela Prieto comentó:
Muy majo

Ilustración 21: Interfaz perfil otro usuario sin iniciar sesión

Esta interfaz tiene un menú en la parte superior idéntico al que se puede encontrar en la interfaz de inicio y la parte central está dividida en dos columnas, en la columna de la izquierda se muestran los datos personales y preferencias del otro usuario a excepción del email y del teléfono, y en la de la derecha se muestra la valoración media que tiene este usuario y los últimos comentarios que le han hecho.

Esta misma interfaz cambia si el usuario ha iniciado sesión y queda como la siguiente:



ShareCar Perfil Contacto Bienvenido David Cerrar sesión

Datos personales

Nombre: *Angela*
Apellidos: *Prieto*
Dirección: *Madrid*
Barrio: *Hortaleza*
Email: *angela@hola.com*
Teléfono: *654999000*

Preferencias

Música: *Normal*
Fumador: *Si*
Hablador: *Mucho*

Valorar:

Puntuación: - +

Valorar

Comentario

Comentar

Valoraciones al usuario:

Puntuación media: 4

Últimos comentarios:

Ilustración 22: Interfaz perfil otro usuario iniciando sesión

Si el usuario ha iniciado sesión hay algunos cambios en esta interfaz, en el menú de la parte superior a la derecha se muestra el mensaje de bienvenida y la opción de cerrar sesión, también, en la parte central en la columna de la izquierda se muestra el email y el teléfono de este usuario, para poder ponerse en contacto con él, y además, en la columna de la derecha se da la opción de puntuar a este usuario y de ponerle un comentario.

Y la última interfaz creada es la de contacto de **ShareCar**:



Ilustración 23: Interfaz contacto ShareCar

Esta interfaz tiene un menú superior, el cual está dividido en el menú de navegación de la aplicación y a la derecha puede tener el formulario para iniciar sesión o registrarse o el mensaje de bienvenida y la opción de cerrar sesión como tiene el menú superior de la interfaz de perfil de usuario.

En la parte central hay un fondo degradado con el nombre de la aplicación y la parte inferior está dividida en tres columnas, en la de la izquierda está la información de contacto de la aplicación **ShareCar**, en la del centro algunos comentarios que han hecho usuarios sobre la aplicación y en la de la derecha hay información sobre la aplicación.

A continuación se expone el diagrama de navegación de la aplicación, hay que indicar que desde cualquier interfaz se puede acceder a las interfaces de inicio, perfil y contacto, esta navegabilidad no se incluye en el diagrama a fin de mejorar la legibilidad del mismo, no sobrecargándolo:

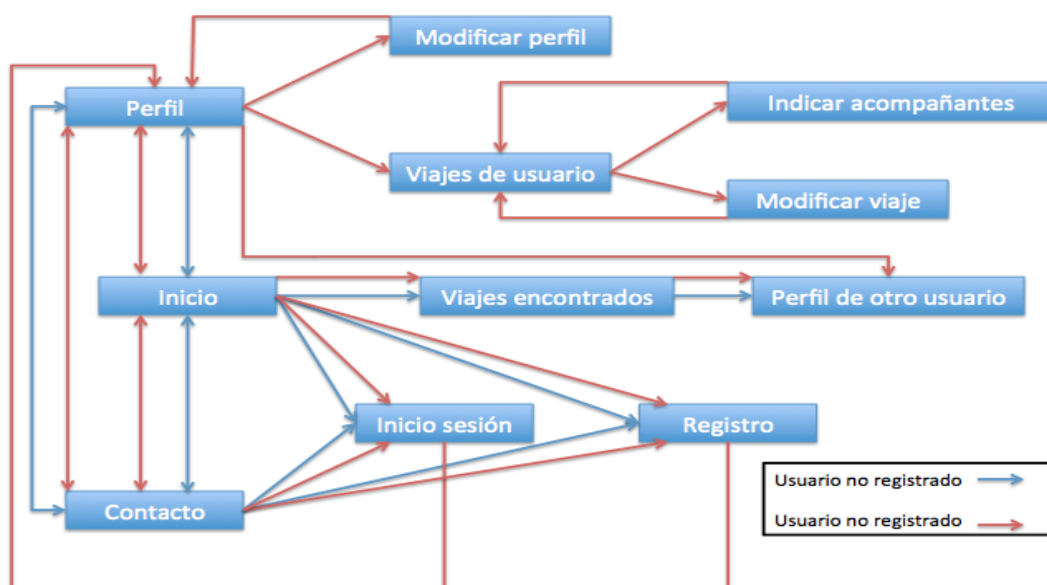


Ilustración 24: Diagrama de navegación

9. Pruebas del sistema

El objetivo de esta sección es definir y determinar las pruebas necesarias mediante las que se pueda comprobar que el sistema cumpla todas las especificaciones, y por lo tanto el correcto funcionamiento del sistema.

Primero se va a definir los tipos de pruebas que se realizarán, los requisitos necesarios para el entorno de pruebas, así como los procedimientos que se van a llevar a cabo en cada una de ellas, asegurando que el sistema cumple las expectativas en las distintas fases del proyecto de manera que se verifique en todo momento el correcto funcionamiento del sistema.

9.1. Especificación del plan de pruebas

Lo primero a tener en cuenta es la especificación de cada una de las pruebas del sistema, es importante saber cuál es el alcance y el entorno de estas pruebas, con el fin de conocer que se va a probar y sobre que se va a probar.

9.1.1. Definición del alcance de las pruebas

En esta sección se especificará el alcance de las pruebas a realizar, es decir, que es lo que será probado en el sistema. En el presente documento por tanto se especifican los siguientes tipos de pruebas:

- **Pruebas de integración:** estas pruebas tendrán la función de probar en conjunto todas las acciones que se pueden llevar a cabo en la aplicación. Detectando todos los posibles errores funcionales.
- **Pruebas de sistema:** estas pruebas consisten en un nivel más de acoplamiento, en este caso todo el sistema. Estas pruebas se realizarán sobre todo el sistema completo, bajo un entorno controlado.
- **Pruebas de implantación:** estas pruebas al igual que las del sistema, testearán el sistema en su totalidad, pero en este caso, sobre un entorno real de ejecución. De esta manera se podrá comprobar el correcto funcionamiento del sistema en su puesta en producción. Teniendo más en cuenta las limitaciones hardware y software relativas a la conexión y el almacenamiento de la aplicación.

9.1.2. Definición de requisitos del entorno de pruebas

Ahora se van a definir los requisitos que tiene que cumplir el entorno de pruebas. Teniendo en cuenta tanto recursos software como hardware. Para su mejor entendimiento se mostrarán con el siguiente formato tabular:

ID	REP-nº	Fuente	Fuente del requisito de pruebas
Nombre	Nombre del requisito de pruebas		
Descripción	Descripción del requisito de pruebas		

Tabla 80: Modelo de tabla REP

9.1.2.1. Software

A continuación se detallan los requisitos del software del entorno de pruebas:

ID	REP-01	Fuente	David Tedejo García
Nombre	Software del servidor		
Descripción	Para la realización de las pruebas se necesitará un servidor con un sistema operativo que incluya un servidor Apache HTTP Server y un gestor de base de datos <i>MySQL Community Edition</i> .		

Tabla 81: REP-01

ID	REP-02	Fuente	David Tedejo García
Nombre	Software de los equipos de trabajo		
Descripción	Los equipos de trabajo deberán de tener el siguiente software: <ul style="list-style-type: none">• Servidor web Apache HTTP Server.• Gestor de base de datos <i>MySQL Community Edition</i>.• Navegador web.		

Tabla 82: REP-02

9.1.2.2. Hardware

Ahora se detallan los requisitos del hardware del entorno de pruebas:

ID	REP-03	Fuente	David Tedejo García
Nombre	Hardware servidor		
Descripción	El servidor deberá de tener las siguientes especificaciones mínimas: <ul style="list-style-type: none">• Procesador de doble núcleo• 2 GB de RAM		

Tabla 83: REP-03

ID	REP-04	Fuente	David Tedejo García
Nombre	Almacenamiento BBDD		
Descripción	Para la realización de las pruebas la base de datos tendrá al menos 2 GB disponibles para el almacenamiento.		

Tabla 84: REP-04

ID	REP-05	Fuente	David Tedejo García
Nombre	Almacenamiento Servidor		
Descripción	Para la realización de las pruebas el servidor tendrá una capacidad de al menos 50 GB de espacio de almacenamiento.		

Tabla 85: REP-05

9.2. Especificación técnica de niveles de pruebas

A continuación incluiremos cada una de las pruebas que se pretenden realizar sobre el sistema para garantizar el debido cumplimiento de los requisitos del sistema.

9.2.1. Pruebas de integración

El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes del sistema con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos especificados en las verificaciones correspondientes.

Para la realización de estas pruebas se utilizará una tabla para normalizar la definición de las diferentes pruebas. Dicha tabla será mostrada a continuación y tendrá los siguientes campos:

Identificador	PI-NNN
Nombre	
Objetivo	
Clase	
Datos de entrada	
Implementación	
Resultado esperado	

Tabla 86: Formato tabla pruebas de integración

- **Identificador:** Identificador unívoco de cada requisito que tendrá el siguiente formato: **PI-NNN**, donde **PI** identifica que es una **Prueba de Integración** y **NNN** es un número de tres cifras diferenciador para cada prueba.
- **Nombre:** Hace referencia a un pequeño título aclarativo y sencillo de lo que trata la prueba.
- **Objetivo:** Objetivo a alcanzar con la realización de la prueba.
- **Clase:** Clases que se usarán para la realización de la prueba.

- **Datos de entrada:** Datos necesarios para la realización de la prueba.
- **Implementación:** Descripción del desarrollo con el que se ejecutaría la prueba.
- **Resultado esperado:** Resultado que se espera tras la realización de la prueba.

Identificador	PI-001
Nombre	Registro de usuario
Objetivo	Un usuario crea una nueva cuenta
Clase	index.php, registro.html, guardarUsuario.php
Datos de entrada	Datos del usuario
Implementación	<ol style="list-style-type: none">1. Abrir navegador web2. Ir a la página de inicio de ShareCar3. Pulsar en Registrarse4. Rellenar todos los campos necesarios5. Pulsar botón Registrarse
Resultado esperado	Se creará un usuario nuevo que pueda acceder al sistema

Tabla 87: Prueba PI-001

Identificador	PI-002
Nombre	Borrado de usuario
Objetivo	Un usuario se da de baja del sistema
Clase	perfil.php, borrarUsuario.php
Datos de entrada	-
Implementación	<ol style="list-style-type: none">1. Abrir navegador web2. Ir a la página de su perfil en ShareCar3. Pulsar en Dar de baja la cuenta
Resultado esperado	Se borrará un usuario del sistema

Tabla 88: Prueba PI-002

Identificador	PI-003
Nombre	Modificación de datos usuario
Objetivo	Un usuario puede modificar sus datos en el sistema
Clase	perfil.php, modificarPerfil.php, guardarUsuario.php
Datos de entrada	Datos de usuario
Implementación	<ol style="list-style-type: none"> 1. Abrir navegador web 2. Acceder a la página de su perfil en ShareCar 3. Pulsar en Modificar datos personales 4. Rellenar los campos que se quieran modificar 5. Pulsar en Guardar cambios
Resultado esperado	Los datos del usuario quedarán modificados en el sistema

Tabla 89: Prueba PI-003

Identificador	PI-004
Nombre	Inicio de sesión
Objetivo	Un usuario inicia sesión en el sistema
Clase	index.php, login.php
Datos de entrada	Email, contraseña
Implementación	<ol style="list-style-type: none"> 1. Abrir navegador web 2. Acceder a ShareCar 3. Introducir usuario 4. Introducir contraseña 5. Pulsar botón Entrar
Resultado esperado	Que el usuario entre al sistema

Tabla 90: Prueba PI-004

Identificador	PI-005
Nombre	Cierre de sesión
Objetivo	Un usuario cierra sesión en el sistema
Clase	cerrarSesion.php
Datos de entrada	Email, contraseña
Implementación	<ol style="list-style-type: none">1. Abrir navegador web2. Acceder a ShareCar3. Introducir usuario4. Introducir contraseña5. Pulsar botón Entrar6. Pulsar botón Cerrar sesión
Resultado esperado	Que la sesión en el sistema finalice

Tabla 91: Prueba PI-005

Identificador	PI-006
Nombre	Creación de viaje
Objetivo	Un usuario puede crear una viaje
Clase	perfil.php, nuevoViaje.php
Datos de entrada	Datos de la nota
Implementación	<ol style="list-style-type: none">1. Abrir navegador web2. Acceder a su perfil en ShareCar3. Rellenar los campos del nuevo viaje4. Pulsar en Crear
Resultado esperado	El viaje será creado y almacenado en el sistema

Tabla 92: Prueba PI-006

Identificador	PI-007
Nombre	Eliminación de viaje
Objetivo	Un usuario puede eliminar un viaje
Clase	perfil.php, viajesUsuario.php, borrarViaje.php
Datos de entrada	-
Implementación	<ol style="list-style-type: none"> 1. Abrir navegador web 2. Acceder a su perfil en ShareCar 3. Pulsar botón Mis viajes 4. Pulsar en Cancelar viaje
Resultado esperado	El viaje será eliminado del sistema

Tabla 93: Prueba PI-007

Identificador	PI-008
Nombre	Modificación de viaje
Objetivo	Un usuario puede modificar su viaje
Clase	perfil.php, viajesUsuario.php, modificarViaje.php, nuevoViaje.php
Datos de entrada	Datos del viaje
Implementación	<ol style="list-style-type: none"> 1. Abrir navegador web 2. Acceder a su perfil en ShareCar 3. Pulsar botón Mis viajes 4. Pulsar en Modificar Viaje 5. Rellenar los datos que se quieran cambiar 6. Pulsar en Modificar
Resultado esperado	Los datos del viaje serán modificados y guardados en el sistema

Tabla 94: Prueba PI-008

Identificador	PI-009
Nombre	Indicar acompañantes viaje
Objetivo	Un usuario puede indicar que otros usuarios han viajado con él
Clase	perfil.php, viajesUsuario.php, viajeros.php, personasViajes.php
Datos de entrada	Email del acompañante
Implementación	<ol style="list-style-type: none"> 1. Abrir navegador web 2. Acceder a su perfil en ShareCar 3. Pulsar botón Mis viajes 4. Pulsar en Indicar acompañantes 5. Introducir el email del acompañante 6. Pulsar en Aceptar
Resultado esperado	El sistema guarda los acompañantes de ese viaje

Tabla 95: Prueba PI-009

Identificador	PI-010
Nombre	Buscar viajes
Objetivo	Un usuario puede buscar viajes
Clase	index.php, viajesEncontrados.php
Datos de entrada	Datos del viaje a buscar
Implementación	<ol style="list-style-type: none"> 1. Abrir navegador web 2. Acceder a ShareCar 3. Rellenar datos del viaje 4. Pulsar botón Buscar
Resultado esperado	Se muestran los viajes que concuerdan con los datos introducidos

Tabla 96: Prueba PI-010

Identificador	PI-011
Nombre	Ver perfil otros usuarios
Objetivo	El usuario puede ver el perfil de otros usuarios
Clase	perfil.php, perfilUsuario.php
Datos de entrada	-
Implementación	<ol style="list-style-type: none"> 1. Abrir navegador web 2. Acceder a su perfil en ShareCar 3. Pulsar Ver perfil de los usuarios que se le recomiendan
Resultado esperado	Ver perfil de otros usuarios

Tabla 97: Prueba PI-011

Identificador	PI-012
Nombre	Puntuar a otros usuarios
Objetivo	El usuario puntúa a otro usuario
Clase	perfil.php, perfilUsuario.php, valorarUsuario.php
Datos de entrada	Puntuación al usuario
Implementación	<ol style="list-style-type: none"> 1. Abrir navegador web 2. Acceder a su perfil en ShareCar 3. Pulsar Ver perfil de los usuarios que se le recomiendan 4. Elegir puntuación 5. Pulsar el Valorar
Resultado esperado	La valoración se guarda en el sistema

Tabla 98: Prueba PI-012

Identificador	PI-013
Nombre	Comentar sobre otro usuario
Objetivo	El usuario pone un comentario a otro usuario
Clase	perfil.php, perfilUsuario.php, comentarUsuario.php
Datos de entrada	Comentario del usuario
Implementación	<ol style="list-style-type: none"> 1. Abrir navegador web 2. Acceder a su perfil en ShareCar 3. Pulsar Ver perfil de los usuarios que se le recomiendan 4. Escribir un comentario 5. Pulsar en Comentar
Resultado esperado	El comentario se guarda en el sistema

Tabla 99: Prueba PI-013

Identificador	PI-014
Nombre	Comentar sobre ShareCar
Objetivo	El usuario pone un comentario sobre ShareCar
Clase	perfil.php, comentarUsuario.php
Datos de entrada	Comentario del usuario
Implementación	<ol style="list-style-type: none"> 1. Abrir navegador web 2. Acceder a su perfil en ShareCar 3. Escribir un comentario sobre ShareCar 4. Pulsar en Comentar
Resultado esperado	El comentario se guarda en el sistema

Tabla 100: Prueba PI-014

Identificador	PI-015
Nombre	Contacto con ShareCar
Objetivo	El usuario ve la información para ponerse en contacto con ShareCar
Clase	contacto.php
Datos de entrada	-
Implementación	1. Abrir navegador web 2. Acceder a ShareCar 3. Pulsar en Contacto
Resultado esperado	Ver información de contacto de ShareCar

Tabla 101: Prueba PI-015

Por último, destacar que **todas las pruebas de integración han tenido un resultado satisfactorio**. Esto se debe a que el resultado esperado y el conseguido han sido el mismo para todas las pruebas descritas anteriormente.

9.2.2. Pruebas de sistema

Estas pruebas se realizan sobre todo el sistema, bajo un entorno controlado. Se han clasificado y separado por los siguientes tipos:

9.2.2.1. Pruebas de rendimiento

El objetivo de estas pruebas es comprobar que todas las acciones que se pueden hacer en la aplicación se realicen en un tiempo de ejecución correcto, sin que el usuario tenga que esperar en ningún momento.

Para la realización de estas pruebas se van a volver a realizar todas las pruebas de integración, comprobando que para ninguna de ellas el tiempo de ejecución es mayor a tres segundos.

Tras la realización de lo descrito en el párrafo anterior se puede concluir que **la aplicación pasa todas las pruebas de rendimiento satisfactoriamente**.

9.2.2.2. Pruebas de facilidad de uso o aceptación

Estas pruebas consisten en comprobar la adaptabilidad del sistema a las necesidades de los usuarios, tanto para asegurar que se acomoda a su modo habitual de trabajo, como para determinar las facilidades que aporta al introducir datos en el sistema y obtener los resultados.

Para la realización de estas pruebas es necesario que el usuario sea quien utilice el sistema para analizar la facilidad de uso.

En el caso de que la facilidad de uso sea escasa, los responsables del diseño del sistema tendrían que rediseñar el sistema (no funcionalidad) para facilitar el uso de la misma.

En este caso, se ha realizado esta prueba con quince usuarios, los cuales no están relacionados con el mundo de la informática y con poca experiencia en el uso de sistemas informáticos, y **todos ellos han aprobado la facilidad de uso de la aplicación**.

9.2.2.3. Pruebas de entorno

Estas pruebas verifican las interacciones del sistema con otros sistemas dentro del mismo entorno. Para la realización de estas pruebas se utiliza una tabla para normalizar la definición de las diferentes pruebas. Dicha tabla se muestra a continuación y tiene los siguientes campos:

Identificador	PSEN-NNN
Nombre	
Objetivo	
Precondiciones	
Implementación	
Resultado esperado	

Tabla 102: Formato tabla pruebas de entorno

- **Identificador:** Identificador unívoco de cada requisito que tendrá el siguiente formato: **PSEN-NNN**, donde **PS** identifica que es una prueba de sistema, **EN** son las dos primeras letras de la categoría de la prueba, en este caso **EN**torno y **NNN** es un número de tres cifras diferenciador para cada prueba.
- **Nombre:** Hace referencia a un pequeño título aclarativo y sencillo de lo que trata la prueba.
- **Objetivo:** Objetivo a alcanzar con la realización de la prueba.
- **Precondiciones:** Condiciones que deben cumplirse para el desarrollo de la prueba.
- **Implementación:** Descripción del desarrollo con el que se ejecutaría la prueba.
- **Resultado esperado:** Resultado que se espera tras la realización de la prueba.

Identificador	PSEN-001
Nombre	Comunicación Apache HTTP Server – Navegador web
Objetivo	Comprobar que Apache HTTP Server y el navegador interactúan de manera correcta para la ejecución de la aplicación
Precondiciones	Que Apache HTTP Server esté arrancado
Implementación	Comprobación de que Apache HTTP Server esté funcionando y que la aplicación se puede ejecutar en cualquier navegador insertando su URL
Resultado esperado	La aplicación se ejecuta en el navegador

Tabla 103: Prueba PSEN-001

Identificador	PSEN-002
Nombre	Comunicación Apache HTTP Server – Servidor MySQL Community Edition
Objetivo	Comprobar que Apache HTTP Server y el servidor MySQL Community Edition se comunican correctamente
Precondiciones	Que Apache HTTP Server y el servidor MySQL Community Edition estén arrancados
Implementación	Comprobación de que existe un intercambio de datos entre los servidores
Resultado esperado	Los dos servidores se comunican de forma correcta

Tabla 104: Prueba PSEN-002

9.2.3. Pruebas de implantación

La función de las pruebas de implantación es la de verificar que la instalación que tiene que realizar el cliente del sistema se realiza correctamente.

En este caso, al tratarse de una aplicación web y dado que la parte servidor de la aplicación se encuentra bajo nuestra responsabilidad, el cliente solo necesitará un navegador web instalado en su máquina.

10. Conclusiones

Las conclusiones se pueden dividir en dos partes, por un lado están las conclusiones que son más personales, es decir, las sensaciones que se han tenido tanto durante el desarrollo del proyecto como al finalizar el mismo, y otras conclusiones más técnicas sobre la aplicación creada.

Tras decidir el proyecto y hablar con el tutor para llevarlo a cabo, lo primero era saber con qué tecnologías iba a desarrollarlo, lo que lleva un trabajo de investigación de las tecnologías actuales y las ventajas e inconvenientes de cada una.

Al final la aplicación se basa principalmente en HTML y PHP, aunque no hay que olvidar que estos lenguajes de programación se complementan con CSS y JavaScript. Esta decisión se debe a que son lenguajes que ya conocía por trabajar con ellos a lo largo del grado y que tienen varias ventajas, como que son lenguajes que son bastante ligeros, baratos a la hora de pensar en comercializar la aplicación y muy eficientes y sencillos.

La parte de programación no me ha resultado muy complicada por los motivos expuesto en el párrafo anterior, aunque sí que me ha llevado su tiempo, porque es una aplicación con parte de servidor y de cliente y había que realizar un diseño eficiente para ella.

La parte de documentación está basada en Métrica V3, ya que he estudiado la especialidad de Sistemas de la información que se basa en gran medida en documentación y dirección de proyectos. Esta parte me ha resultado más complicada que la programación pero el resultado ha sido satisfactorio, ya que se ha desarrollado un prototipo completo y funcional acorde a las especificaciones definidas.

Desde un punto de vista menos personal y más técnico, la aplicación cumple con lo esperado en todo momento, ya que su funcionamiento es óptimo, su diseño es sencillo y fácil de usar, es multiplataforma, es decir, se puede usar en varios dispositivos y navegadores sin problemas, y es bastante ligera y eficiente.

Por último, destacar que se ha realizado un especial énfasis en desarrollar una documentación muy completa, cubriendo todos los requisitos para un proyecto de ésta magnitud. Cumple en todo momento con los estándares seguidos y es coherente para las distintas fases del proyecto.

11. Líneas futuras

Esta sección se centra en el futuro que puede tener la aplicación creada y cuáles pueden ser los siguientes pasos que se deberían llevar si se sigue adelante con ella. Esta aplicación tiene competidores ya en el mercado, como se puede ver en la sección 3. Estudio del estado del arte.

Pero, por otra parte, si se quiere poner en el mercado podría tener éxito si se realiza una buena publicidad, ya que presenta elementos disruptivos comparado con las aplicaciones similares, como puede ser la recomendación de perfiles similares al propio, algo que podría dar la superioridad necesaria para poder competir en el mercado.

Los siguientes pasos que se podrían llevar a cabo antes de poner la aplicación en el mercado sería realizar algunas mejoras como puede ser hacer la aplicación multilenguaje y extenderla a toda España, ya que de momento al ser la versión beta está hecha sólo para Madrid.

Líneas futuras posteriores podrían consistir en la ampliación de funcionalidades proporcionadas en el sistema en base a encuestas realizadas a los usuarios (necesidades específicas, *feedback*), así como desarrollar aplicaciones cliente específicas para los principales sistemas operativos móviles. También se pretende realizar una difusión de la plataforma una vez implantada a través de las principales redes sociales y buscadores web.

12. Referencias

A continuación se muestran las páginas web dónde se ha buscado y recogido información para el desarrollo del proyecto, tanto en la codificación como en la documentación:

[1] Ministerio de Administraciones Públicas. (2001). *Arquitectura Métrica Versión 3*. Consultado entre Marzo y Junio de 2014. http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#U6XhDo1 tro

[2] Real Academia Española. (2001). *Diccionario de la Real Academia Española*. Consultado entre Marzo y Junio de 2014. <http://www.rae.es/recursos/diccionarios/drae>

[3] Fundación Wikipedia. (2001). *Términos buscado en Wikipedia*. Consultado entre Marzo y Junio de 2014. <http://es.wikipedia.org/wiki/Wikipedia:Portada>

[4] Apache Group. (1995). *Apache HTTP Server*. Consultado entre Abril y Junio de 2014. <https://httpd.apache.org/>

[5] Oracle. (2009). *MySQL Community Edition*. Consultado entre Abril y Junio de 2014. <http://www.mysql.com/products/community/>

[6] Mark, Otto & Jacob. (2011). *BootStrap*. Consultado entre Abril y Junio de 2014. <http://getbootstrap.com/>

[7] Mazella, F., Nappez, F. & Brusson, N. (2004) *Página web de BlaBlaCar*. Consultado entre Abril y Junio de 2014. <http://www.blablacar.es/>

[8] Reinicke, M., Weber, S. & Siedler, M. (2001). *Página web de Carpooling*. Consultado entre Abril y Junio de 2014. <http://www.carpooling.es/>

[9] Hidalgo Demeusois, D. & Figueres, J.M. (2009). *Página web de Amovens*. Consultado entre Abril y Junio de 2014. <https://www.amovens.com/es/>

[10] Google. (2008). *Navegador Google Chrome*. Consultado entre Abril y Junio de 2014. <http://www.google.es/intl/es/chrome/browser/>

[11] Apple Inc. (2003). *Navegador Safari*. Consultado entre Abril y Junio de 2014. <https://www.apple.com/es/safari/>

[12] Corporación Mozilla. (2004). *Navegador Mozilla Firefox*. Consultado entre Abril y Junio de 2014. <http://www.mozilla.org/es-ES/firefox/new/>

[13] Nixon, R. (2012). *Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites* (2ª ed.). Estados Unidos de América: Andy Oram

13. Anexo I. Presupuesto

En esta sección se explican los aspectos económicos que afectan al proyecto. Para lograr este fin, se da comienzo con el presupuesto y se prosigue con la explicación del entorno socio-económico actual, sobre el que se han basado gran cantidad de las decisiones tomadas.

13.1. Presupuesto

Como se puede apreciar en la siguiente sección (que aborda el tema de la planificación), la duración del proyecto ha sido de 101 días aproximadamente, con una media de trabajo de cinco horas diarias (total: **505 horas**). A lo largo de este tiempo han surgido una serie de gastos que ha sido necesario cubrir. Las estimaciones sobre los mismos se exponen a continuación.

13.1.1. Personal

Para la realización del proyecto se ha necesitado cubrir las tareas específicas que corresponden a un analista de sistemas, un arquitecto de *software*, un diseñador, un programador y un experto en calidad y pruebas. Todas estas tareas han sido realizadas por el alumno, cambiando éste de rol según fuese necesario. Además, ha sido necesario un tutor que ha realizado gran parte del trabajo de jefe de proyecto, proporcionando asistencia en los momentos necesarios.

A continuación, se muestra una estimación de los costes que generan cada uno de esos roles por hora de trabajo:

Personal	
Cargo/Rol	Coste (por hora)
Analista	20,00 €
Arquitecto	15,00 €
Diseñador	15,00 €
Gestión de calidad y pruebas	20,00 €
Jefe de proyecto	30,00 €
Programador	10,00 €

Tabla 105: Estimación de coste de personal por hora

Ahora aplicamos estos costes desglosando las horas de trabajo para cada actividad:



Actividades y recursos							
Actividad	Personal (horas)						Totales (horas)
	Jefe de proyecto	Analista	Arquitecto	Diseñador	Pruebas	Programador	
Motivación y objetivos	3	3	3	1	0	0	10
Planteamiento del problema	6	10	4	0	0	0	20
Valoración del estado del arte	4	6	2	0	0	0	12
Realización de la comparativa	1	2	1	0	0	0	4
Objetivos de la aplicación	1	2	1	0	0	0	4
Elección de características del sistema	6	5	8	6	0	5	30
Metodología a utilizar	2	2	3	2	0	2	14
Arquitectura a sistema	2	1	3	2	0	1	13
Entorno tecnológico del sistema	2	2	2	2	0	2	10
Análisis del sistema	14	30	1	0	0	0	45
Alcance del sistema	2	4	1	0	0	0	7
Identificación de usuarios	2	6	0	0	0	0	8



Actividades y recursos							
Actividad	Personal (horas)						Totales (horas)
	Jefe de proyecto	Analista	Arquitecto	Diseñador	Pruebas	Programador	
Especificación requisitos <i>software</i>	5	10	0	0	0	0	15
Especificación casos de uso	5	10	0	0	0	0	15
Diseño del sistema	13	0	4	30	0	3	50
Definición de la arquitectura del sistema	5	0	2	4	0	0	11
Requisitos de diseño	4	0	0	10	0	0	14
Diseño de clases	2	0	1	10	0	2	15
Diseño físico de datos	2	0	1	6	0	1	10
Implementación del prototipo	7	0	3	5	0	20	35
Estructura del prototipo	3	0	3	3	0	7	16
Diseño e implementación de las interfaces del prototipo	4	0	0	2	0	13	19
Pruebas del sistema	5	0	0	0	30	0	35

Actividades y recursos							
Actividad	Personal (horas)						Totales (horas)
	Jefe de proyecto	Analista	Arquitecto	Diseñador	Pruebas	Programador	
Especificación del plan de pruebas	2	0	0	0	12	0	14
Especificación técnicas de niveles de pruebas	3	0	0	0	18	0	21
Presupuesto	5	10	0	0	0	0	15
Planificación	7	3	2	1	0	2	15
Planificación inicial	3	1	1	1	0	1	7
Planificación final	3	1	1	0	0	1	6
Comparativa planificaciones	1	1	0	0	0	0	2
Revisión del proyecto	15	5	0	0	0	0	20
Horas totales	139	114	47	85	60	60	505
Coste	4170,00 €	2280,00 €	705,00 €	1275,00 €	1200,00 €	600,00 €	10.230,00 €

Tabla 106: Coste asociado a los recursos y horas empleadas

13.1.2. Material

El coste derivado del uso del material puede ser dividido en dos apartados:

- Dispositivos *hardware* empleados a lo largo del proyecto.
- Licencias de *software* necesarias para llevarlo a cabo.

13.1.2.1. Hardware

Para estimar el coste de los equipos se ha realizado el cálculo de la amortización de su precio respecto al tiempo que han sido usados en base al coste supuesto en el momento de adquisición. Se ha estimado una vida útil de unos tres años.

Como no tiene sentido realizar una amortización en función del número de días, se aproximarán los meses trabajados a tres y medio (101 días \approx 3,36 meses).

En la siguiente tabla se muestra el coste derivado del uso de los equipos:

Hardware					
Nombre	Precio unitario	Unidades	Coste (Ud./mes)	Nº meses	Total
MacBook Pro i5	1.229,00 €	1	34,14 €	3,5	119, 49 €
Total					119, 49 €

Tabla 107: Coste asociado al *hardware* empleado

13.1.2.2. Software

Se ha intentado abaratar al máximo en cuanto a coste por lo que a *software* se refiere. Para ello se han empleado el máximo posible de aplicaciones gratuitas. Los costes por licencia del *software* más significativo se desglosan en la siguiente tabla:

Software			
Nombre	Nº Licencias	Coste/licencia	Coste
Mac OS X Mavericks	1	0,00 €	0,00 €
Windows 8	1	0,00 €	0,00 €
Microsoft Office	1	119,00 €	119,00 €
Apache HTTP Server	1	0,00 €	0,00 €
MySQL Community Edition	1	0,00 €	0,00 €
Total			119,00 €

Tabla 108: Coste asociado al software empleado

13.1.3. Costes indirectos

En esta sección se incluyen los costes del consumo de luz y agua, el servicio de limpieza, la red de telefonía y comunicaciones necesarias y un seguro a todo riesgo durante los tres meses y medio de duración del proyecto. Como se ha trabajado desde el hogar no se incluye coste por alquiler de oficina.

Costes Indirectos	
Nombre	Coste
Luz, agua	175,00 €
Servicio de limpieza	210,00 €
Teléfono fijo y red de comunicaciones	105,00 €
Seguro a todo riesgo	35,00 €
Total	525,00 €

Tabla 109: Costes indirectos

13.1.4. Resumen de costes

Calculando la suma económica necesaria para cubrir los costes derivados de los apartados anteriores, se necesitan 10.443,49 euros para realizar el proyecto. Sin embargo, esta cantidad no incluye el impuesto sobre valor añadido ni los márgenes de beneficio y riesgo asociados a este tipo de proyectos.

Resumen de costes	
Nombre	Cantidad
Personal	10.230,00 €
Hardware	119,49 €
Software	119,00 €
Costes indirectos	525,00 €
Total	10.993,49 €

Tabla 110: Resumen de costes

13.1.5. Coste total

Llegados a este punto se puede obtener el precio final necesario para llevar a cabo este proyecto. Se muestra en la siguiente tabla:

Presupuesto total	
Nombre	Cantidad
Coste total	10.993,49 €
Riesgo (5% del coste total estimado)	549,68 €
Beneficio (20% del coste total estimado)	2.198,70 €
Total sin I.V.A	13.741,87 €
I.V.A. (21%)	2.885,80 €
Total	16.627,67 €

Tabla 111: Presupuesto total

Teniendo en cuenta todos los presupuestos calculados anteriormente las ganancias ascienden a **2.198,70 euros**, y el precio total del proyecto a **16.627,67 euros**.

14. Anexo II. Planificación

En esta sección se muestra la planificación realizada del proyecto (explicándose para ello tanto la **planificación inicial** como la **planificación final**), así como el método que se sigue para realizar el seguimiento del proyecto y evitar desviaciones. Se realiza una **comparación** entre ambas planificaciones (estimada y real).

Para elaborar la planificación del proyecto se ha utilizado la herramienta *Microsoft Project 2013* en Windows 8, no se ha incluido en el presupuesto ya que se ha utilizado una versión de prueba gratuita. El diagrama de Gantt ha sido generado mediante dicha herramienta.

14.1. Planificación inicial

Para la realización de la planificación inicial del proyecto se utiliza el siguiente cronograma que muestra las distintas etapas del proyecto, con sus respectivas actividades y el tiempo que deberá ocupar cada una de estas.

Planificación inicial			
Nombre de tarea	Duración	Comienzo	Fin
Motivación y objetivos	5 días	lun 03/03/14	vie 07/03/14
Planteamiento del problema	7 días	lun 10/03/14	mar 18/03/14
Valoración del estado del arte	3 días	lun 10/03/14	mié 12/03/14
Realización de la comparativa	2 días	jue 13/03/14	vie 14/03/14
Objetivos de la aplicación	2 días	lun 17/03/14	mar 18/03/14
Elección de características del sistema	9 días	lun 24/03/14	jue 03/04/14
Metodología a utilizar	4 días	lun 24/03/14	jue 27/03/14
Arquitectura del sistema	3 días	vie 28/03/14	mar 01/04/14
Entorno tecnológico del sistema	2 días	mié 02/04/14	jue 03/04/14

Planificación inicial			
Nombre de tarea	Duración	Comienzo	Fin
Análisis del sistema	13 días	lun 07/04/14	mié 23/04/14
Alcance del sistema	2 días	lun 07/04/14	mar 08/04/14
Identificación de usuarios	2 días	mié 09/04/14	jue 10/04/14
Especificación requisitos software	5 días	vie 11/04/14	jue 17/04/14
Especificación caso de uso	3 días	lun 21/04/14	mié 23/04/14
Diseño del sistema	14 días	jue 24/04/14	mar 13/05/14
Definición de la arquitectura del sistema	2 días	jue 24/04/14	vie 25/04/14
Requisitos de diseño	3 días	lun 28/04/14	mié 30/04/14
Diseño de clases	5 días	lun 05/05/14	vie 09/05/14
Diseño físico de datos	4 días	jue 08/05/14	mar 13/05/14
Implementación del prototipo	8 días	mié 14/05/14	vie 23/05/14
Estructura del prototipo	3 días	mié 14/05/14	vie 16/05/14
Diseño e implementación de las interfaces del prototipo	5 días	lun 19/05/14	vie 23/05/14
Pruebas del sistema	8 días	lun 26/05/14	mié 04/06/14
Especificación del plan de pruebas	3 días	lun 26/05/14	mié 28/05/14

Planificación inicial			
Nombre de tarea	Duración	Comienzo	Fin
Especificación técnica de niveles de pruebas	5 días	jue 29/05/14	mié 04/06/14
Presupuesto	4 días	lun 24/03/14	jue 27/03/14
Planificación	58 días (8 días reales)	mié 19/03/14	vie 06/06/14
Planificación inicial	3 días	mié 19/03/14	vie 21/03/14
Planificación final	3 días	lun 02/06/14	mié 04/06/14
Comparativa planificaciones	2 días	jue 05/06/14	vie 06/06/14
Revisión del proyecto	6 días	lun 09/06/14	lun 16/06/14
Total	78 días	lun 03/03/14	lun 16/06/14

Tabla 112: Planificación inicial

Se puede extraer como conclusión lo siguiente, se estima que el proyecto durará 101 días, y se aproxima que **78 días** se habrán dedicado al trabajo en el proyecto. Se estima una media de cinco horas por día.

14.2. Diagrama de Gantt inicial

A continuación se muestra el diagrama de Gantt resultante de la planificación inicial anterior, se ha dividido en tres imágenes para que se puedan apreciar los detalles con mayor claridad.

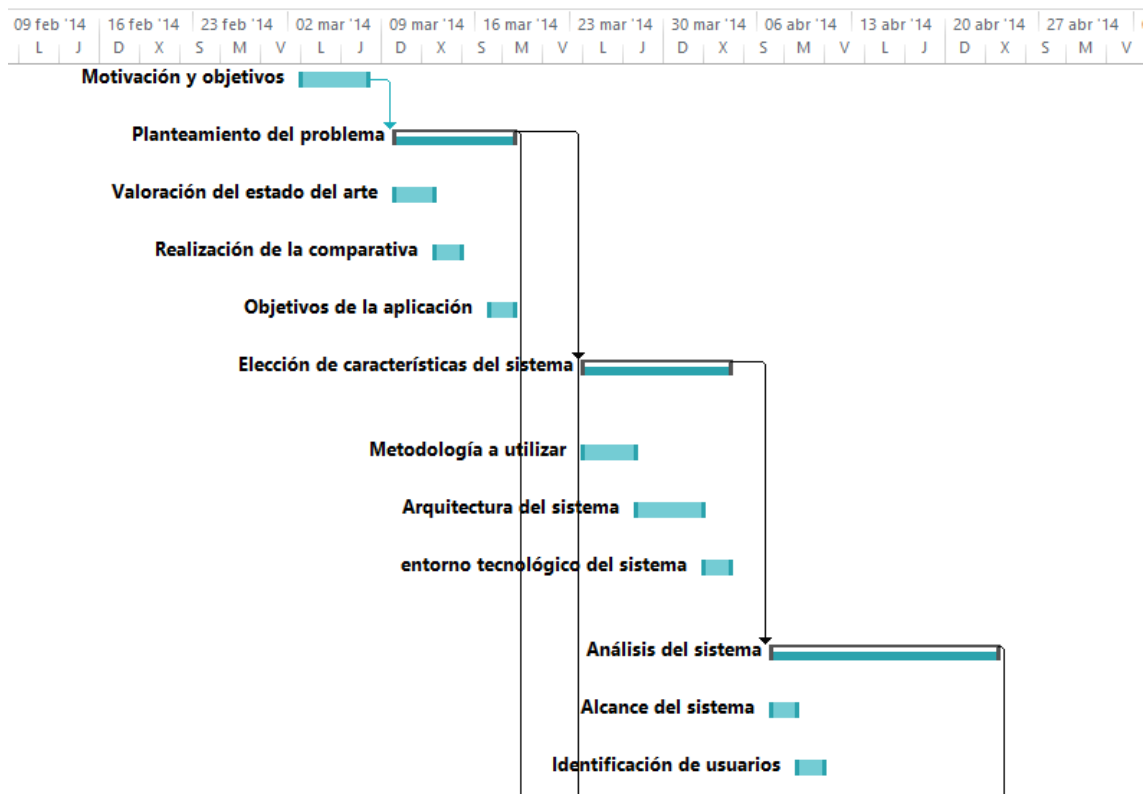


Ilustración 25: Diagrama de Gantt inicial (1)

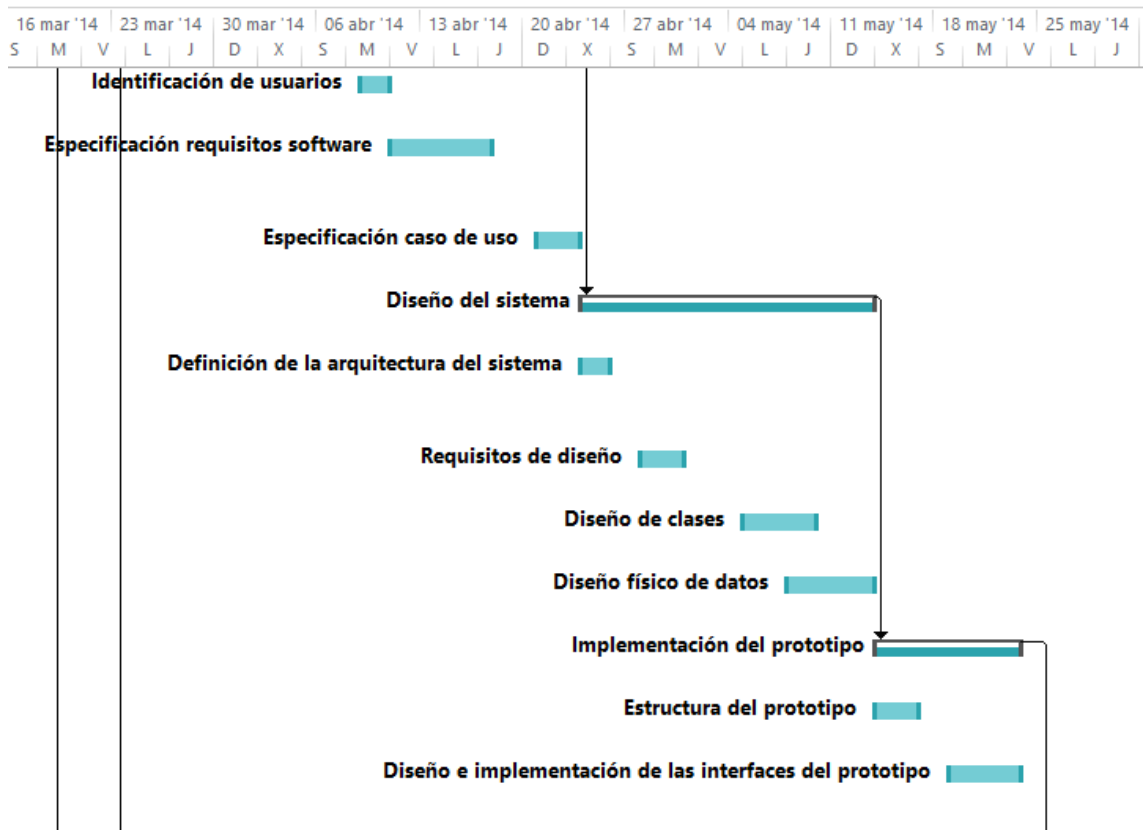


Ilustración 26: Diagrama de Gantt inicial (2)

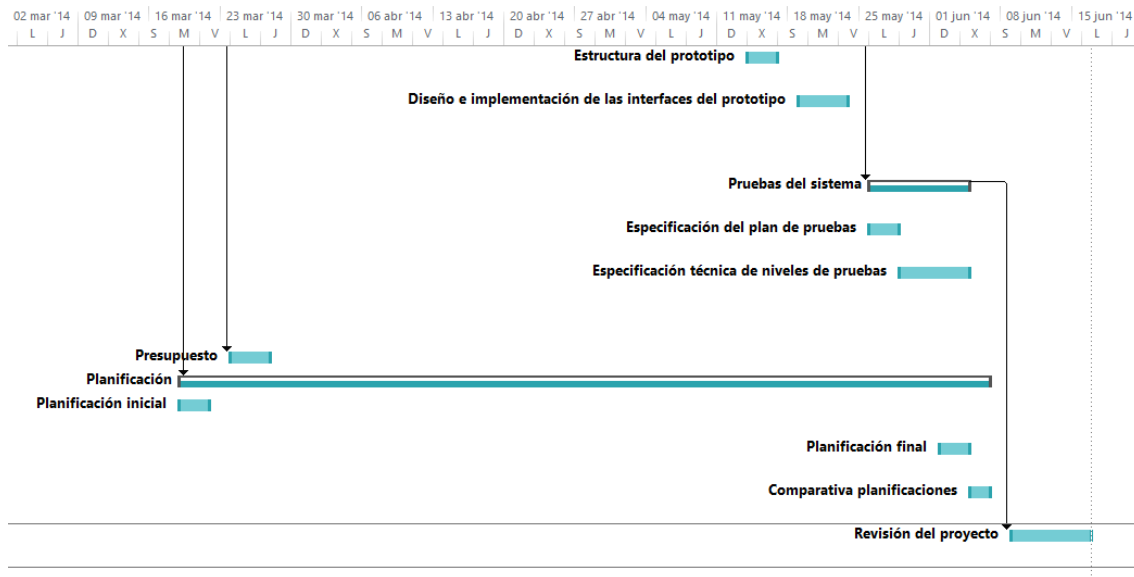


Ilustración 27: Diagrama de Gantt inicial (3)

14.3. Planificación final

Para un correcto seguimiento y control del proyecto, por lo general, se convocaron reuniones de manera asidua, procurando realizarse una por semana o una por cada dos semanas. De esta forma, se pudo realizar un seguimiento sobre el correcto avance del proyecto, así como si se estaban alcanzando los objetivos fijados.

El control sobre el proyecto se realizó mediante el siguiente cronograma que muestra las distintas etapas del proyecto, con sus respectivas actividades y el tiempo que necesitó cada una de estas.

La única fecha crítica que se debía tener en cuenta es la entrega del trabajo, como máximo, el día 22 de junio de 2014.

Planificación final			
Nombre de tarea	Duración	Comienzo	Fin
Motivación y objetivos	5 días	mar 04/03/14	lun 10/03/14
Planteamiento del problema	6 días	mié 12/03/14	mié 19/03/14
Valoración del estado del arte	3 días	mié 12/03/14	vie 14/03/14
Realización de la comparativa	2 días	lun 17/03/14	mar 18/03/14

Planificación final			
Nombre de tarea	Duración	Comienzo	Fin
Objetivos de la aplicación	1 día	mié 19/03/14	mié 19/03/14
Elección de características del sistema	11 días	lun 24/03/14	lun 07/04/14
Metodología a utilizar	5 días	lun 24/03/14	vie 28/03/14
Arquitectura del sistema	4 días	lun 31/03/14	jue 03/04/14
Entorno tecnológico del sistema	2 días	vie 04/04/14	lun 07/04/14
Análisis del sistema	16 días	lun 07/04/14	lun 28/04/14
Alcance del sistema	2 días	lun 07/04/14	mar 08/04/14
Identificación de usuarios	2 días	mié 09/04/14	jue 10/04/14
Especificación requisitos software	7 días	vie 11/04/14	lun 21/04/14
Especificación caso de uso	5 días	mar 22/04/14	lun 28/04/14
Diseño del sistema	17 días	mar 29/04/14	mié 21/05/14
Definición de la arquitectura del sistema	3 días	mar 29/04/14	jue 01/05/14
Requisitos de diseño	3 días	lun 05/05/14	mié 07/05/14
Diseño de clases	6 días	mié 07/05/14	mié 14/05/14
Diseño físico de datos	5 días	jue 15/05/14	mié 21/05/14

Planificación final			
Nombre de tarea	Duración	Comienzo	Fin
Implementación del prototipo	10 días	mié 21/05/14	mar 03/06/14
Estructura del prototipo	3 días	mié 21/05/14	vie 23/05/14
Diseño e implementación de las interfaces del prototipo	8 días	vie 23/05/14	mar 03/06/14
Pruebas del sistema	7 días	mar 03/06/14	mié 11/06/14
Especificación del plan de pruebas	3 días	lun 26/05/14	mié 28/05/14
Especificación técnica de niveles de pruebas	4 días	jue 29/05/14	mar 03/06/14
Presupuesto	3 días	mié 26/03/14	vie 28/03/14
Planificación	59 días	lun 24/03/14	jue 12/06/14
Planificación inicial	2 días	lun 24/03/14	mar 25/03/14
Planificación final	2 días	mié 11/06/14	jue 12/06/14
Comparativa planificaciones	1 día	jue 12/06/14	jue 12/06/14
Revisión del proyecto	6 días	vie 13/06/14	vie 20/06/14
Total	86 días	mar 04/03/14	vie 20/06/14

Tabla 113: Planificación final

Se puede extraer como conclusión lo siguiente, la duración del proyecto ha sido de 101 días, y se aproxima que **86 días** se habrán dedicado al trabajo en el proyecto. Se ha trabajado aproximadamente unas cinco horas por día (de media).

14.4. Diagrama de Gantt final

A continuación se muestra el diagrama de Gantt resultante de la planificación final anterior, se ha dividido en tres imágenes para que se puedan apreciar los detalles con mayor claridad.

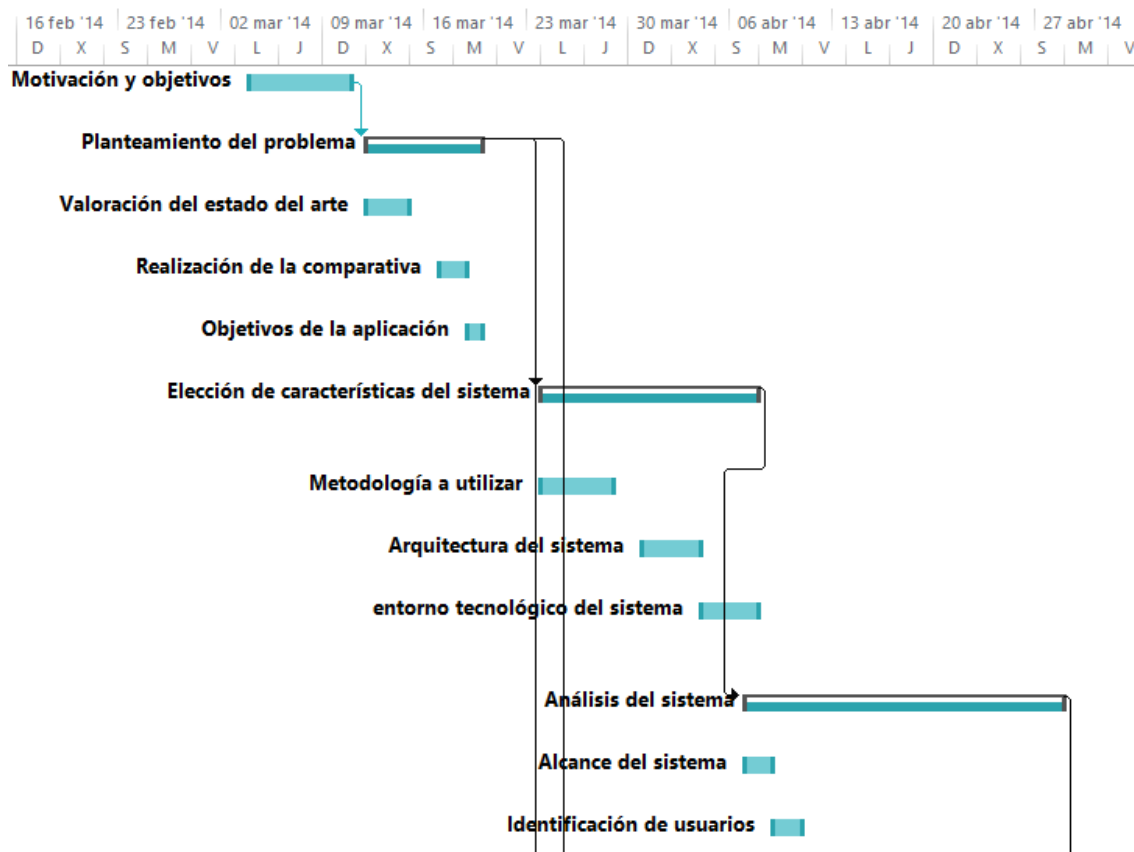


Ilustración 28: Diagrama de Gantt final (1)

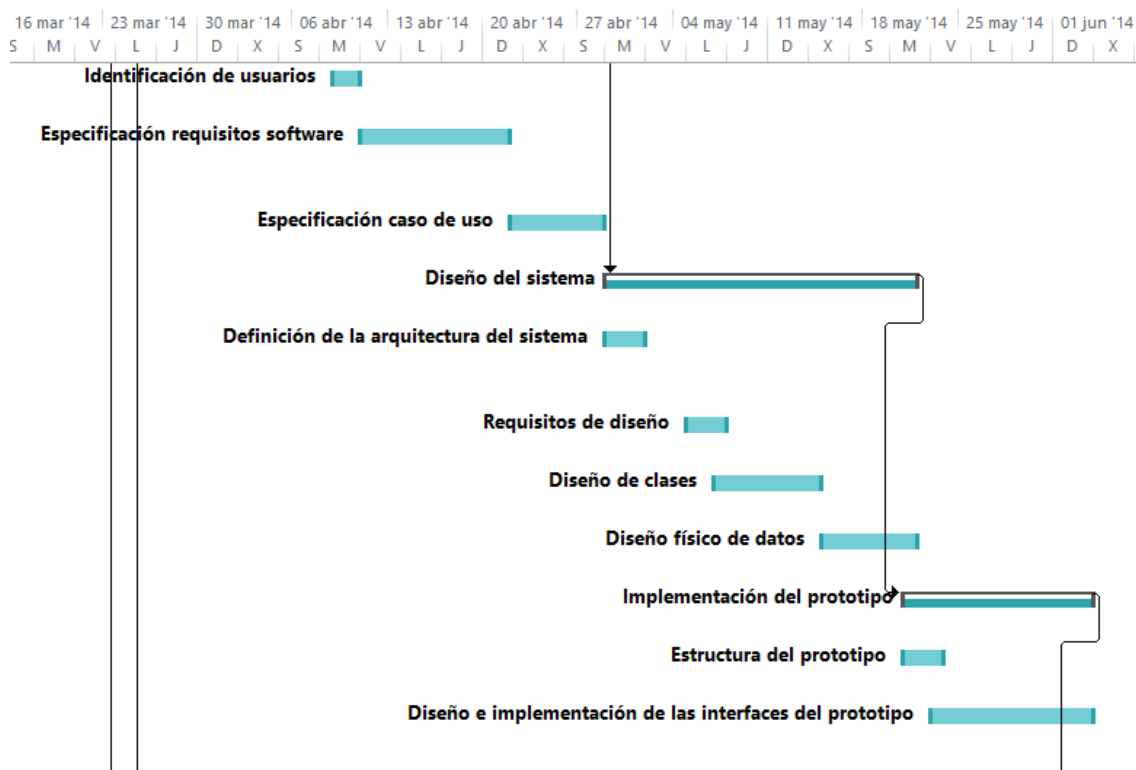
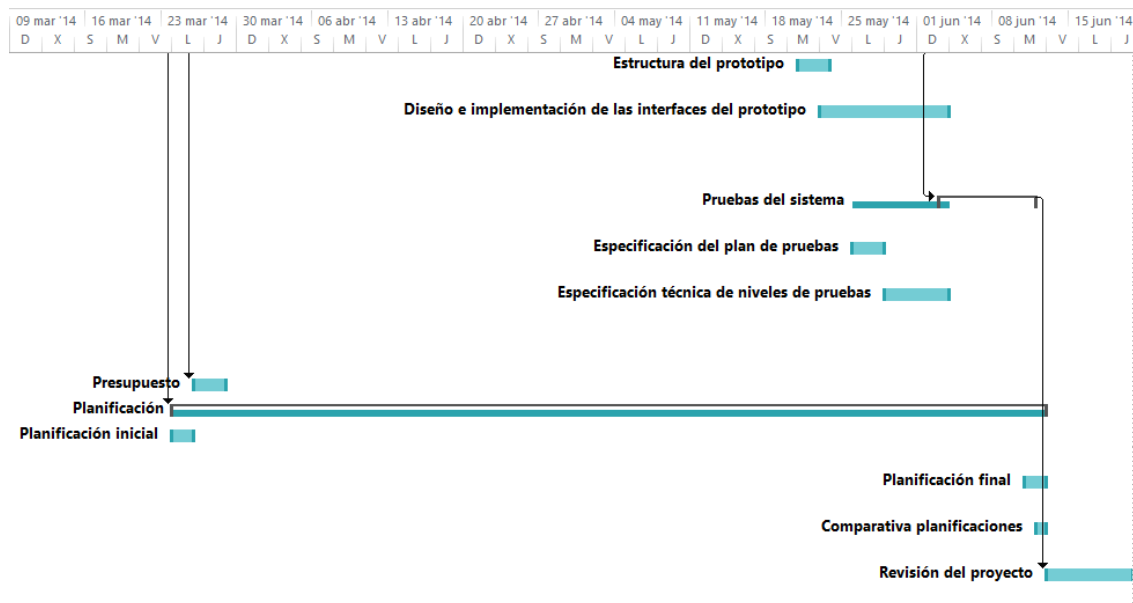


Ilustración 29: Diagrama de Gantt final (2)



14.5. Comparativa planificaciones

No existe ninguna diferencia significativa entre ambas planificaciones. La planificación final implica mayor tiempo para la realización del proyecto, por lo que la desviación es negativa, es decir, se ha invertido más tiempo real que el planificado inicialmente. Siempre se han tomado decisiones con el objetivo de encontrar un equilibrio entre la cantidad de trabajo que se puede realizar en el proyecto y el tiempo que es necesario para cumplir con éxito ese trabajo planteado.

Con el desarrollo de la planificación inicial y de la final, se busca comprender la **evolución** que ha sufrido el proyecto a lo largo del tiempo, y descubrir si esta planificación ha sido correcta para poder terminar exitosamente un proyecto de esta envergadura con el tiempo necesario.